VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

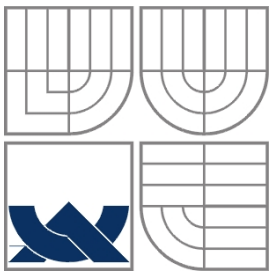
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

GEOLOKACE
URČENÍ LOKALITY POMOCÍ IP ADRESY

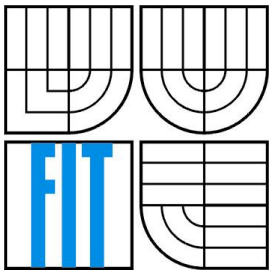
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE **Radovan Zvonček**
AUTHOR

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

GEOLOKACE
URČENÍ LOKALITY POMOCÍ IP ADRESY

GEOLOCATION - DETERMINING LOCALITY FROM THE IP ADDRESS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **Radovan Zvonček**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. Adam Herout, Ph.D.**
SUPERVISOR

BRNO 2008

Abstrakt

Práce řeší problém geolokace jako procesu, jehož úkolem je mapovat IP adresu na zeměpisnou polohu. Ta může být vyjádřena zeměpisnými souřadnicemi nebo specifikaci země, regionu a města. Právě druhou možností se zabývá tato práce.
Nejprve vysvětluje pojem a popisuje dostupné služby. Na základě získaných poznatků popisuje návrh a implementaci vlastního systému. Navrhované řešení testuje na vzorcích adres. Dosažené výsledky jsou stabilní pro zemi, ale poměrně nepřesné při odhadu města

Abstract

Geolocation is a process which tries to determine geographical location of a device connected to the internet. Location can be represented by exact coordinates or by description of country, region and city, which is what this thesis is focused on.
First, it explains the concept of geolocation and introduces available solutions. Next, it describes the design and implementation of own solution on the basis of gathered information. Solution is tested on two sets of IP addresses. Gathered results are relatively correct when determining countries, but contain inaccuracies when determining cities.

Klíčová slova

IP adresa, geografická poloha, geolokácia, algoritmus

Keywords

IP address, geographic location, geolocation, algorithm

Citace

Radovan Zvonček: Geolokace - určení lokality pomocí IP adresy, bakalářská práce, Brno, FIT VUT v Brně, rok 2009

Geolokace - určení lokality pomocí IP adresy

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Adam Herouta, Ph.D.
Další informace mi poskytl Ing. Jiří Vrba, Ph.D.
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Radovan Zvonček
18.5.2009

Poděkování

Rád by som vyjadril svoju vďaku doktorovi Vrbovi za to, že ma opakovane upozorňoval na odklonenie od témy, ktorého som pomerne často dopúšťal, a za ústretovosť a dostupnosť pri dohadovaní termínov konzultácii.
Nemenej rád by som poďakoval doktorovi Heroutovi za cenné rady toľko potrebné pri formálnych úpravách práce.

Obsah

1 Úvod

Mojou úlohou v tejto práci je zoznámiť sa s pojmom geolokácie a spôsobmi akými je v súčasnom internete implementovaná. Po preskúmaní nevyhnutných teoretických základov a štúdiu dostupných produktov poskytujúcich geolokačné služby sa pokúsim navrhnúť vlastné riešenie a porovnať jeho výsledky s dostupnými produktmi.

Práca je teda zameraná na návrh ako celok a dôraz na samotné určenie polohy nie je až taký veľký.

V kapitole *O probléme* geolokácie sa venujem popisu geolokácie ako postupu pre určenie geografickej polohy. Ďalšia podkapitola je venovaná popisu niektorých existujúcich služieb a produktov. V poslednej podkapitole sa pokúšam popísať konkrétne algoritmy používané týmito službami.

Kapitola *Návrh vlastného algoritmu* najskôr popisuje pre túto prácu zaujímavú geolokáciu z IP adresy. V nasledujúcich podkapitolách presne vymedzím, čo je úlohou môjho návrhu a popíšem vstupy a ich štruktúru. Po tom budem môcť popísať samotný algoritmus spolu s jeho výstupom.

Nasledujúca kapitola *Implementácia* obsahuje popis celého systému počnúc rozhraním, až po spôsob uloženia dát.. Popísal som v nej aj použité technológie a dôvody, ktoré ma k tomu viedli.

Predposledná kapitola obsahuje ukážku testov vykonávaných na dvoch vzorkách adries: užívateľmi overené adresy a náhodne vybrané adresy univerzít. Dosiahnuté výsledky porovnávam s výsledkami ostatných služieb a produktov a snažím sa vysvetliť nezrovnalosti.

Poslednou kapitolou je *Záver*, v ktorom stručne zhodnotím a uzavriem túto prácu.

2 Geolokácia

V tejto kapitole by som rád všeobecne popísal a vysvetlil pojem geolokácia. Uvediem príklady použitia a výhody z toho vyplývajúce. Tiež sa pokúsim ukázať niektoré negatívne aspekty s geolokáciou spojené. V poslednej podkapitole uvediem príklady niektorých (ne)komerčných produktov.

2.1 O geolokácii

Proces určenia geografickej polohy zariadenia pripojeného do internetu sa nazýva geolokácia. Za zariadenie je možné považovať počítač, mobilný telefón, smerovač alebo akékoľvek iné zariadenie, ktoré obsahuje sieťové rozhranie a je možné mu priradiť internetovú adresu.

Geolokačné algoritmy môžu využívať radu vstupných dát – od IP adries, cez GPS údaje a metadata v obrázkoch, až po nastavenia jazyku a časovej zóny v konkrétnom zariadení. V mojej práci sa budem zaoberať výhradne IP adresami a k ním pridruženými informáciami, dostupnými v DNS a `whois` databázach. Ak v ďalšom texte spomeniem geolokáciu a nekonkretizujem ináč, budem mať na mysli práve zisťovanie polohy zariadenia na základe IP adresy.

Samotná lokalizácia sa môže chápať ako presne určenie geografických súradníc, alebo ako množina atribútov, ktoré človeku poskytnú zrozumiteľnejšiu informáciu o polohe zariadenia. Ak sa človek dozvie, že zariadenie je v meste Brno nachádzajúcom sa v Jihomoravskom kraji Českej republiky, dozvie sa toho podstatne viac, než keby mu bola poskytnutá dvojica čísel 49.2, 16.633.

Preto sa v tejto práci venujem prvému spôsobu.

Výsledky týchto algoritmov bývajú zhromažďované v databázach a užívateľ k ním pristupuje pomocou dotazov. Takýto prístup je efektívny – výsledok jedného vykonania lokalizačného algoritmu môže byť poskytnutý viacerým užívateľom. Za nevýhodu sa dá považovať nutnosť preventívne aktualizovať uložené informácie.

Takéto databázy majú spravidla charakter komerčných produktov. Jednak je potrebné zaplatiť za poskytnutie dát v nich obsiahnutých, jednak sú samotné algoritmy predmetmi obchodného tajomstva a nie je možné sa k ním dostať. Existujú aj nekomerčné databázy, tie však neposkytujú natoľko presné alebo aktuálne informácie. Podrobne sa existujúcimi službami a produktmi budem zaoberať v jednej z nasledujúcich podkapitol.

2.2 Výhody a nevýhody

Využitie geolokácie a výhody z nej plynúce sa najnázornejšie demonštrujú na príkladoch. Internetové vyhľadávače dokážu skrátiť čas strávený samotným hľadaním a vrátiť presnejší výsledok ak poznajú geografickú polohu svojho klienta.

Komunikačné aplikácie a protokoly (Skype, Jabber,...) môžu informovať užívateľov nielen o stave svojich kontaktov, ale aj o časovom pásme, v ktorom sa nachádzajú alebo aktuálnom počasí v regióne kontaktu.

Nezanedbateľné je aj marketingové využitie, ktoré by umožnilo spoločnostiam presnejšie cieliť reklamu a ponúkať lokálne žiadané produkty.

V neposlednom rade sa geolokácia dá využiť v rámci webových serverov. Ak by server poznal polohu svojho klienta pripojeného z krajiny netotožnej s umiestnením servera, mohol by mu automaticky poskytnúť stránku v príslušnom jazyku. Rovnako dobre by vedel užívateľovi odoprieť obsah stránok, ktorý je v danej krajine nelegálny.

Takáto činnosť sa už dá však považovať za nežiaducu. Koncept internetu spočíva vo vytvorení globálnej siete, pre ktorú neplatia geografické hranice a sloboda prejavu je jedna z najvyšších hodnôt. Akékoľvek filtrovanie obsahu je teda neprípustné.

Ďalej, prispôsobovanie jazyku webových stránok nemusí byť vždy prospešné. Európsky obchodník na obchodnej ceste v Číne nebude príliš nadšený z čínskeho textu svojich obľúbených internetových stránok.

Napokon, informácia o polohe užívateľa sa dá považovať za osobný údaj a ako každý osobný údaj môže byť aj zneužitá. Geolokácia nie je exaktný proces a jej výsledok sa takmer nikdy nedá považovať za úplne spoľahlivý.

2.3 Existujúce služby a produkty

Geolokáciou sa zaoberá niekoľko komerčných ale aj voľne dostupných produktov. Pred tým ako popíšem niektoré z nich, uvediem ako je geolokácia začleňovaná do týchto produktov. Narazil som na dva prístupy:

- Produkt sa zaoberá výhradne geolokáciou. Takéto produkty sa špecializujú na vyhľadanie čo najviac informácii o príslušnej adrese. Produkt tohto typu je napríklad *IP2location* [5].
- Produkt je iba súčasťou balíka viacerých služieb. Typickým príkladom je *Geobytes* [6]. Geolokácia je iba jedna zo služieb produktu, preto na ňu nie je kladený taký dôraz a poskytuje skromnejšie odpovede.

Popis produktov obsahuje iba základné informácie. Popis metód, ktoré používajú, je uvedený v kapitole 2.4. Výsledky, ktoré dosahujú, rovnako ako ich porovnanie, sa nachádzajú v kapitole 5.

2.3.1 NetGeo

NetGeo bol vytvorený a spravovaný organizáciou CAIDA [4]. Zdrojové súbory neboli verejne dostupné, avšak je možné nájsť popis algoritmu, ktorý tento produkt používa. Tento algoritmus voľne popíšem v kapitole 2.4.1.

Systém sa skladá z databázy a kolekcie skriptov napísaných prevažne v jazyku Perl. Rozhranie pre interakciu so serverom umožňuje jednoduchú integráciu do webových stránok.

NetGeo sa snaží mapovať IP adresy a čísla autonómnych systémov (tj. množiny IP adries a smerovačov pod spoločnou technickou správou) na geografickú polohu. Dokáže určiť zemepisnú šírku a dĺžku, štát, kraj a mesto.

Databáza *NetGeo* prestala byť aktualizovaná pred niekoľkými rokmi, takže odpovede ním poskytnuté sú zastarané nepresné.

2.3.2 IP2location

Ide o komerčný produkt, čo znamená, že technické údaje o postupoch získavania dát alebo ich uložení nie sú dostupné.

Produkt poskytuje o skúmanej IP adrese viacero informácii – štandardné údaje o krajine a meste, ale aj exotickejšie číslo telefónnej predvoľby pre zistenú krajinu alebo kód lokálnej meteorologickej stanice.

K dispozícii je niekoľko variantov produktu z hľadiska typu poskytnutých informácii, počtu rozhraní a prostredí, v ktorých je možné produkt používať.

Stránky [5] umožňujú bezplatné vyskúšanie limitované na niekoľko adries denne.

Je pravdepodobné, že tento produkt disponuje viacerými verziami databáz čo do aktuálnosti. Demonštračné príklady teda môžu vykonávať dotazy nad staršími databázami, pričom reálnym klientom je poskytnutá aktuálna databáza.

Na základe týchto pozorovaní si dovolím špekulovať. Pri klientskom (demonštračnom alebo platenom) dotaze je použitá výhradne databáza a je vrátený výsledok v nej uložený. V prípade neúspechu sa nevykoná samotný algoritmus lokalizácie adresy; maximálne si skúmanú adresu zapamätá. Odhadujem, že v mesačných intervaloch sa systém dávkovo pokúsi lokalizovať takto označené adresy. Je možné, že skontroluje aj uložené adresy, ktoré sú síce v korektné a úplné, ale mohlo dôjsť k zmene pôvodných údajov – napríklad zmena vlastníka adresy.

IP Address	Country	Region	City	Latitude/ Longitude	ZIP Code	Time Zone	
147.229.220.177	CZECH REPUBLIC	JIHOMORAVSKY KRAJ	BRNO	49.2 16.633	-	+01:00	Map It
	Net Speed	ISP				Domain	
	DSL	BRNO UNIVERSITY OF TECHNOLOGY				VUTBR.CZ	
	IDD Code	Area Code	Weather Station				
	420	-	EZXX0002 - BRNO				

Obrázok 2.1: Ukážka IP2location – webové rozhranie

2.3.3 GeoIP

GeoIP je názov technológie vlastnenej firmou MaxMind, Inc. Podľa domovskej stránky produktu [7], je založená na neinvazívnom vyhľadávaní a určovaní geografických a iných informácii určených pre firemných zákazníkov v reálnom čase. Rozsah typov poskytnutých informácii je porovnateľný s *IP2location*.

Na rozdiel od *IP2location* používa informácie od užívateľov, ktorí prostredníctvom webových stránok zadávajú informácie o svojej polohe. Tieto informácie, ktorých počet sa údajne pohybuje v miliónoch, spracované a mapované na konkrétne IP adresy.

K dispozícii je demo prístupné cez domovskú stránku produktu.

Hostname	Country Code	Country Name	Region	Region Name	City	Postal Code	Latitude	Longitude	ISP	Organization
147.229.220.177	CZ	Czech Republic	78	Jihomoravsky Kraj	Brno		49.2000	16.6333	Brno University of Technology	Brno University of Technology

Obrázok 2.2: Ukážka GeoIP – webové demo

2.3.4 Geobytes

Geobytes je názov skupiny balíkov pre prispôsobenie obsahu internetových stránok [6]. Ide o komerčný produkt, ktorý je všetkými svojimi aspektmi prispôsobený na čo najjednoduchšie použitie pri tvorbe internetových stránok.

Balíky poskytujú rôzne funkcie ako napríklad štatistiky prezentujúce počet návštevníkov stránky zoskupených podľa miesta pôvodu alebo prispôsobené vkladanie uvítacieho textu.

Nie všetky balíky sú však platené – napríklad balík *GeoDirection* je dostupný zdarma a jeho úlohou je presmerovať užívateľov na základe krajiny pôvodu.

2.4 Existujúce algoritmy

Počas skúmania problematiky som sa pokúšal získať a porovnať konkrétne algoritmy, ktoré jednotlivé produkty používajú. Zistil som však, že to nie je také jednoduché.

Väčšina algoritmov je proprietárna, takže akékoľvek informácie o nich sú nedostupné a o týchto algoritmoch sa dá iba špekulovať.

Ďalšia skupina je síce dostupná verejne, ale na jej sprístupnenie je potrebné zaplatiť určitý registračný poplatok.

Napokon sú tu dva algoritmy. Jeden je používaný produktom *NetGeo* a uvediem jeho voľný popis. Druhý má názov *Systems and methods for determining collecting and using geographic locations of internet users*, je dostupný na [1] a pokúsim sa ho detailnejšie analyzovať.

2.4.1 NetGeo algoritmus

Tento algoritmus nie je na stránkach *NetGeo* žiadnym spôsobom špecifikovaný, dostupný je pomerne stručný slovný popis. Ten sa dá zhrnúť do nasledujúcich bodov:

1) Vstupom algoritmu je skúmaná IP adresa
2) Prvým krokom je dotaz na databázu, ktorého účelom je zistiť, či už bola požadovaná adresa niekedy skúmaná.
3) Ďalší krok spočíva v vykonaní a analyzovaní jedného alebo viacerých dotazov na databázu `whois`. Na analýzu sú použité skripty v jazyku Perl. Bližší popis týchto skriptov sa mi nepodarilo získať.
4) V prípade, že `whois` neposkytne dostatočné informácie, *NetGeo* sa pokúsi analyzovať telefónne číslo, aby tak spresnil vrátený výsledok. Spôsob, akým to uskutoční sa mi však tiež nepodarilo zistiť.
5) Výstupom algoritmu sú v predošlej podkapitole spomenuté údaje: zemepisná šírka a dĺžka, štát, kraj a mesto.

Dostupný popis je veľmi všeobecný a vynecháva tie dôležité a zaujímavé časti, ktorými je napríklad samotná analýza `whois` odpovedí.

Užitočným aspektom je rozdelenie na databázu a výkonné skripty. Motiváciou je snaha zabrániť opakovanému vykonávaniu skriptov pri duplicitnom dotaze na jednu IP adresu, pretože za vykonaním skriptov sa skrýva potenciálna časová náročnosť. Ide však o takmer intuitívne rozdelenie a každý slušný geolokačný algoritmus by s ním mal pracovať.

2.4.2 Systems and methods for determining collecting and using geographic locations of internet users

Ide o patent obsahujúci popis robustného systému pre určovanie pozícií užívateľov pripojených do internetu. Aby bolo možné prečítať celý dokument a nie len abstrakt, je potrebné sa registrovať na príslušnej stránke.

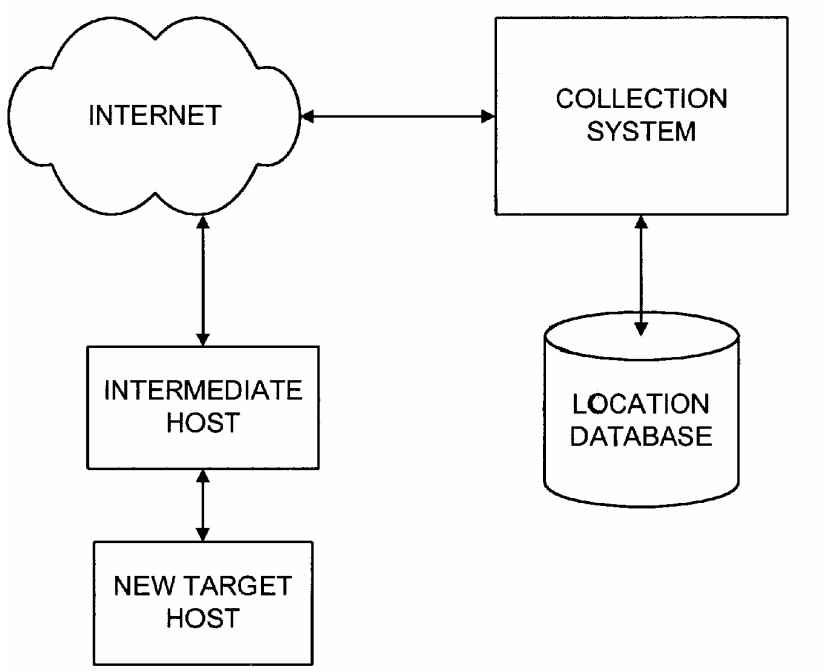
Riešenie sa nezaoberá iba skúmaním IP adries a k ním príbuzným informácií, ale využíva aj iné zdroje (údaje od poskytovateľa pripojenia do internetu) a postupy.

Patent popisuje napríklad spôsob zavedenia systému do praktického použitia ako súčasť webového serveru, alebo spôsob, akým kontrolovať, kto o zistené informácie žiada a prípadne mu odoprieť prístup. V tejto práci uvediem iba tie časti popísaného patentu, ktoré súvisia s geolokáciou podľa IP adresy.

K dispozícii je pomerne podrobná špecifikácia, vrátane vývojových diagramov a ukážok manipulácie s výstupmi služieb `whois` a `traceroute`.

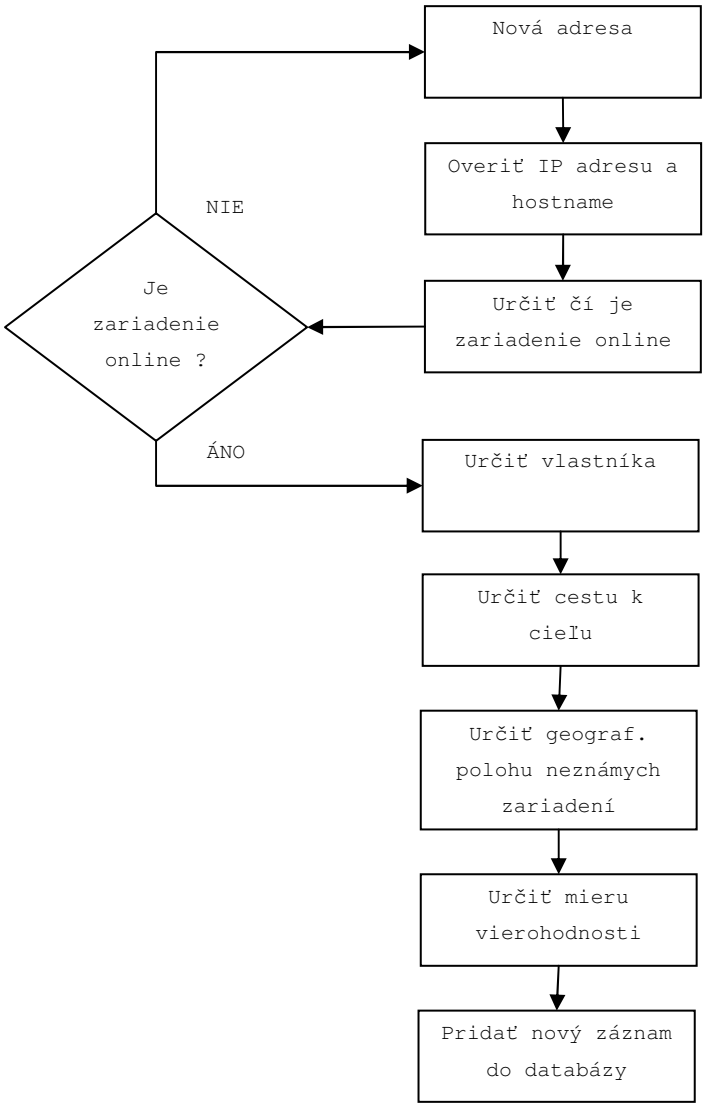
2.4.2.1 Schéma riešenia

- Oddelená výkonná časť Systém zbierania lokácii a databázu
- Súčasťou systému sú okrem skúmaného zariadenia aj prechodné zariadenia, ktoré sú použité pri procese určovania polohy



Obrázok 2.3: Schéma riešenia prevzatá z dokumentu [1]

2.4.2.2 Algoritmus spracovania novej IP adresy



Obrázok 2.4: Zisťovanie polohy

- Overenie IP adresy je vykonané pomocou utility `nslookup`. Účelom je zistiť, či sú údaje korektné.
- Pre test dostupnosti zariadenia sa používa utilita `ping`. V prípade nedostupnosti môže zaradiť adresu do fronty adries čakajúcich na ďalšie spracovanie.
- Za vlastníka zariadenia je považovaný majiteľ IP adresy skúmaného zariadenia. Pre získanie potrebných informácii sa používa utilita `whois`.
- Cestou k cieľu sa rozumie výstup utility `traceroute` spustenej na zariadení, na ktorom pracuje popisovaný systém smerom k skúmanej adrese

- Výstup utility `traceroute` poskytne zoznam prechodných zariadení. Záznamy o týchto zariadeniach vo sú vyhľadávané v databáze. Ak ich nenájde, pokúsi sa okrem cieľového zariadenia určiť polohu aj týchto zariadení.
- Získané informácie ohodnotí mierou vierohodnosti v rozsahu od 0 (najmenej) až 100 a získané dáta uloží do databázy.

2.4.2.3 Ostatné zaujímavé vlastnosti

Napriek množstvu informácií, ktoré patent obsahuje, niektoré dôležité časti neuvádza. Patrí sem napríklad podrobnejší opis spôsobu, akým analyzovať odpoveď na `whois` dotaz. Tento problém nie je jednoduchý a budem sa mu venovať v jednej z nasledujúcich kapitol.

Príklady výstupov použitých utilít `whois` a `treceroute` sa mi nepodarilo reprodukovať. Patent bol zverejnený v roku 2004 a je pravdepodobné, že sa za uplynutý čas zmenila organizácia dát v príslušnej `whois` databáze. Reprodukcia `treceroute` stroskotala na niektorom z prechodných smerovačov, ktorý začal filtrovať príslušné pakety a mne sa teda nepodarilo získať úplnú cestu k uvedenému zariadeniu.

Určenie vierohodnosti

Pre každú analyzovanú adresu je evidovaný atribút vierohodnosť. Tento atribút určuje nakoľko je získaná informácia spoľahlivá. Na hodnotu atribútu má vplyv viacero faktorov:

- Prechodné zariadenia, ktoré sú často používané – napríklad smerovače blízko zariadeniu, na ktorom tento systém beží – sú analyzované opakovane a ak pri každej ďalšej analýze vrátia rovnaký výsledok, ich vierohodnosť narastie.
- Zariadenie, ktoré síce nie je uložené v databáze a jeho poloha je neznáma, ale nachádza sa v susedstve známych zariadení, automaticky získa vysokú vierohodnosť.
- Ak má zariadenie vierohodnosť 99, znamená to, že jeho poloha bola potvrdená treťou stranou. Tou môže byť napríklad analytik spravujúci systém.
- Vierohodnosť 100 môže zariadenie dostať až po overení vlastným majiteľom zariadenia. Najčastejšie ide o poskytovateľov internetu, ktorých zariadenia sa často používajú ako prechodné.

Konvencie pomenovávania zariadení

Niektorý poskytovatelia internetu majú vo zvyku nastavovať hostname svojich zariadení tak, aby odzrkadľoval ich geografickú polohu.

Napríklad je pravdepodobné, že zariadenie s hostname p1-0-0.sanjose-br2-bbnplanet.net sa naozaj nachádza v San Jose, California, alebo aspoň používa také zariadenie na pripojenie do internetu.

Informácie získané od ISP

Už som spomenul, že sú použité dáta získane od poskytovateľa pripojenia. Robí to tak, že má k dispozícii účet u daného poskytovateľa – v podstate je jeho klientom.

Systém dostane od poskytovateľa pridelenú adresu. Na základe tejto adresy vykoná analýzu pseudonáhodného zariadenia spravovaného poskytovateľom. To je však motivované získaním prechodných zariadení pod správou poskytovateľa a možnosťou ich analýzy. Poloha analyzovaných zariadení môže byť potvrdená samotným poskytovateľom, a tak poskytnúť užitočné oporné body.

Takéto operácie však môžu byť náročné na kapacitu linky, preto sú spravidla vykonávané v časoch mimo špičky.

3 Návrh vlastného algoritmu

V tejto kapitole popíšem návrh vlastného algoritmu, ktorého úlohou bude podľa IP adresy zariadenia pripojeného do internetu zistiť jeho geografickú polohu, ktorá bude reprezentovaná údajmi o krajine, meste a prípadne kraji, v ktorom sa zariadenie nachádza.

V predošlom texte som síce povedal, že budem hovoriť o geolokácii podľa IP adresy. Tú som však podrobnejšie neopísal. Preto tejto téme venujem prvú podkapitolu.

Skôr ako budem môcť zrozumiteľne opísať samotný algoritmus, musím zdroje dát, z ktorých bude algoritmus čerpať.

Ďalšie podkapitoly už budú zamerané na popis samotného algoritmu.

3.1 Geolokácia z IP adresy

Táto metóda geolokácie sa snaží určiť polohu zariadenia výhradne na základe jeho IP adresy. Geografickú polohu sa snaží odhadnúť pomocou porovnania skúmanej adresy so známou adresou iného susediaceho zariadenia (serveru, smerovaču...). Okrem toho čerpá z databáz, ktoré nie sú primárne určené na tento účel, ale poskytujú doplňujúce informácie o adrese.

Vychádza z faktu, že prideľovanie IP adries nie je náhodné a existuje v ňom organizácia. Preto dokáže väčšina metód spoľahlivo určiť aspoň krajinu, v ktorej sa zariadenie nachádza. Čím presnejšia sa však snaží byť, tým väčšie nepresnosti môžu vzniknúť. Dôvodom sú napríklad celoštátni poskytovatelia pripojenia do internetu, ktorí majú klientov rozložených po celom štáte.

Na spresnenie je potrebné získať ďalšie informácie. Tu je možné využiť informácie o doméne, z ktorej skúmaná IP adresa pochádza.

Ďalšou možnosťou je získať údaje o správcovi adresného priestoru, z ktorého adresa pochádza. Tiež je možné využiť smerovacie informácie.

Výsledok je teda získaný kombináciou viacerých metód a vstupov.

Z formálnej stránky problému, existujú návrhy RFC dokumentov snažiace sa o definíciu štandardu pre zdroje a celkovú infraštruktúru pre celý proces. Tieto návrhy sa však nestretli s veľkou podporou verejnosti.

3.2 Zdroje dát

Aby bol navrhovaný algoritmus použiteľný a poskytoval aspoň trochu vierohodné výsledky, musí existovať spôsob, akým môže o skúmanej adrese zistiť doplňujúce informácie.

Najlepším spôsobom je pokúsiť sa zistiť, aké informácie o adrese poskytuje príslušná `whois` databáza.

Môže nastať aj situácia, kedy nebude možné zadanú IP analyzovať. V takom prípade je vhodné aby mal algoritmus k dispozícii spôsob, ktorým by dokázal získať adresy susedných zariadení a pokúsil sa ich použiť ako náhradu za chýbajúcu adresu. Na tento účel je vhodná utilita `traceroute`.

3.2.1 Whois

Pojmom `whois` sa označuje protokol založený na princípe dotaz/odpoveď medzi klientom a oficiálnou databázou. Oficiálna databáza obsahuje informácie o IP adresách používaných v internete.

Pre mňa najzaujímavejšie sú údaje o vlastníkoch a správcoch IP adries. Je bežné, že jeden subjekt vlastní viacero adries. V tomto prípade sa hovorí o bloku adries alebo o adresnom rozsahu.

3.2.1.1 O protokole

Protokol nebol navrhnutý s vyhliadkou automatizácie. Odpovede poskytované databázou sú primárne určené pre ľudského čitateľa, nie pre stroj. Z tohto dôvodu neexistuje žiaden štandard, ktorý by definoval formát a význam uložených informácii. Existuje len niekoľko RFC dokumentov (napr.[8]) , ktoré obsahujú súbor smerníc a odporučení.

Ďalej, oficiálna databáza je rozdelená na niekoľko častí nazvaných podľa organizácií, ktoré ich spravujú:

- ARIN – Spravuje adresy v Severnej Amerike
- RIPE NCC - Spravuje európske adresy
- APNIC – Spravuje Ázijské adresy a adresy Austrálie a Oceánie
- LACNIC – Spravuje adresy Južnej Ameriky a karibskej oblasti
- AfriNIC – Spravuje adresy v Afrike

Každá z týchto databáz navyše poskytuje odlišný formát odpovede, čo ešte viac sťažuje automatizáciu manipulácie s týmito údajmi.

3.2.1.2 Štruktúra odpovedí

Napriek týmto skutočnostiam je v odpovediach na dotazy možné nájsť istú organizáciu. Tá síce nie je úplne aplikovateľná na všetky organizácie súčasne, ale základné mechanizmy aspoň čiastočne umožňujú automatizáciu spracovania odpovedí.

Odpoveď je tvorená sekvenciou riadkov. Hoci sa presný formát riadku sa líši medzi jednotlivými organizáciami, je možné určiť všeobecný formát riadku:

```
Názov_položky<oddeľovač>hodnota
```

Ak riadok začína špeciálnym znakom, napríklad % alebo #, zvyšok riadku obsahuje komentár.

Niektoré organizácie (napr. RIPE), poskytujú odpovede, ktoré majú riadky zoskupené do blokov. Dva bloky sú navzájom oddelené aspoň jedným prázdnym riadkom. Prvý riadok bloku je možné označiť za identifikátor bloku a ostatné položky za jeho atribúty.

Typická odpoveď z databázy RIPE môže vyzerať napríklad takto:

```
[Querying whois.arin.net]
[Redirected to whois.ripe.net:43]
[Querying whois.ripe.net]
[whois.ripe.net]
% This is the RIPE Whois query server #3.
% The objects are in RPSL format.
%
% The RIPE Database is subject to Terms and Conditions.
% See http://www.ripe.net/db/support/db-terms-conditions.pdf

% Note: This output has been filtered.
%       To receive output for a database update, use the "-B" flag.

% Information related to '147.229.0.0 - 147.229.255.255'

inetnum:        147.229.0.0 - 147.229.255.255
netname:        VUTBR-TCZ
descr:          Brno University of Technology
descr:          Brno
country:        CZ
admin-c:        VS47
tech-c:         VZ36-RIPE
status:         ASSIGNED PI
mnt-by:         TENCZ-MNT
remarks:        Please report network abuse -> abuse@vutbr.cz
source:         RIPE # Filtered
```

Niektoré bloky bývajú v rámci jednej odpovede jedinečné, niektoré sa opakujú a vzťahujú sa inému, spravidla jedinečnému, bloku.

Iné organizácie (napr. ARIN), poskytujú jediný blok ako odpoveď. Kompenzujú to však zreteľnejším formátom riadku a štandardizovanejším formátom poskytnutých informácii. Uvediem jeden príklad:

```
xzvonc00@merlin: ~$ whois 68.201.145.103
[Querying whois.arin.net]
[Redirected to ipmt.rr.com:4321]
[Querying ipmt.rr.com]
[ipmt.rr.com]
%rwhois  V-1.5:003fff:00  cdptpa-ipmt-rwhois-01.cdptpa.rr.com  (by  Network
Solutions, Inc. V-1.5.9.6)
network:Class-Name:network
network:ID:NETBLK-ISRR-68.201.128.0/17
network:Auth-Area:68.201.128.0/17
network:Network-Name:ISRR-68.201.128.0
network:IP-Network:68.201.128.0/17
network:IP-Network-Block:68.201.128.0 - 68.201.255.255
network:Organization;I:Road Runner
network:Tech-Contact;I:ipaddreg@rr.com
network:Admin-Contact;I:IPADD-ARIN
network:Created:20090514
network:Updated:20090514
network:Updated-By:ipaddreg@rr.com
```

Detailnejšie sa analýze odpovedí získaných z `whois` databáz budem venovať v podkapitole 3.3.4.

3.2.2 Traceroute

`Traceroute` je utilita, ktorá je súčasťou každého väčšieho operačného systému. Používa sa na zistenie prechodných zariadení medzi dvoma počítačmi.

Pracuje na princípe rozposielania packetov s inkrementujúcim sa TTL atribútom, čím získava informácie o stále vzdialenejších smerovačoch.

Výstup poskytuje viacero informácii, z ktorých nie všetky tú pre mňa relevantné. Nepotrebujem poznať čas, ktorý ubehol kým bol príslušný packet prenesený medzi zariadeniami. Stačí mi, že viem, že packet úspešne dorazil. Súčasťou výstupu sú aj doménové mená prechodných zariadení. Tie môžu byť užitočné, ale môj algoritmus s nimi nepracuje.

Negatívom tejto utility je, že niektoré smerovače môžu takéto packety filtrovať a utilita sa stane nepoužiteľnou. Pri nesprávnom použití môže utilita pracovať príliš dlho, čo v prvom rade spomaľuje celý algoritmus. Tiež sa však môže stať, že bude pracovať až tak dlho, že dôjde k vypršaniu časového limitu a nebude vrátený žiaden užitočný výsledok.

3.3 Popis samotného riešenia

Nasledujúce kapitoly sa budú zaoberať popisom navrhnutého algoritmu. Začnú systémom ako celkom a ak to bude nutné. Komplikovanejšie kroky popíšu podrobnejšie.

3.3.1 Hlavný algoritmus

Činnosť celého algoritmu je možné zovšeobecniť do nasledujúcich štyroch krokov:

- Vstupom algoritmu je IP adresa zariadenia, ktorého polohu sa má pokúsiť určiť
  1. Ak ide o privátnu alebo nepoužiteľnú adresu, skončí
  2. Zistí, či zadná adresa nepatrí do už preskúmaného adresného rozsahu. Ak nájde zhodu a analýza nebola vykonaná príliš dávno, vráti tento výsledok a skonči.
  3. Analyzuje IP
  4. Uloží alebo aktualizuje výsledok
- Výstupom algoritmu je adresný rozsah, do ktorého adresa patrí a údaje o krajine, meste a regióne

Za privátne a nepoužiteľné adresy považujem neverejné adresy a adresy pre multicast ako boli špecifikované v dokumente RFC 3330 [12]. Obmedzil som ich však na tieto adresy:

- 10.0.0.0/8, 127.0.0.0/8, 192.168.0.0/16
- multicast 224. - 239.

V kroku 2 nie je dôležité ako dlhý časový úsek uplynul od poslednej analýzy zadanej adresy. Údaje vo `whois` databázach, z ktorých pochádza väčšina informácii, sa nemenia príliš často, stať sa to však môže. Preto je vhodné, aby boli získané informácie z času na čas overené.

Uloženie alebo aktualizácia výsledku sa môže vykonať dvoma spôsobmi. Ak analýza IP prebehne v poriadku a algoritmus vráti vierohodné informácie, je adresa spolu s jej údajmi uložená medzi známe adresy.

Ak však analýza prebehne chybne, alebo vráti nedôveryhodný výsledok, zaradí sa IP adresa spolu s prípadnými určenými údajmi medzi skupinu adries, ktoré sú klasifikované ako nepreskúmané.

Tretí krok, analýza IP, je komplexnejší a zaslúži si vlastnú podkapitolu.

3.3.2 Analýza IP

Analýzou IP mám na mysli jednotlivé kroky, ktoré je nutné vykonať, aby algoritmus získal potrebné informácie. Algoritmus postupuje nasledovne:

1. Získa a spracuje informácie získané z `whois` databázy pre zadanú adresu
2. Ak dostane úplný a spoľahlivý výsledok, môže skončiť.
3. V prípade, že nie sú dostupné všetky informácie, vykoná sa `traceroute` k skúmanej adrese.
4. Nastaví príznak nedôveryhodnosti výsledku, pretože skúmanie adries takto získaných môže byť úplne nepresné.
5. Adresy získané týmto spôsobom sú použité pre doplnenie informácii.

Použitie utility `traceroute` je v tomto algoritme priamočiare. Slúži výhradne na získanie adries prechodných zariadení. Skrýva sa v ňom však potenciál, ktorý v skratke popíšem v podkapitole o vylepšeniach, ktoré som v rámci tejto práce už neimplementoval.

3.3.3 Využitie výstupu traceroute

Výstupom utility je množina adries zariadení, cez ktoré putujú pakety medzi zariadením, na ktorom je vykonávaný tento algoritmus a skúmaným zariadením.

Zariadenia sú organizované sekvenčne a platí, že čím skôr sa zariadenie vyskytuje vo výstupe, tým bližšie sa nachádza k lokálnemu zariadeniu.

Posledná položka patrí buď skúmanému zariadeniu alebo jeho bezprostrednému susedovi.

Algoritmus preto spracováva tento výstup od konca a snaží sa analyzovať jednotlivé IP adresy podobne ako v kroku 1 v predchádzajúcej podkapitole. Skončí, keď doplní všetky informácie alebo vyčerpá všetky adresy.

3.3.4 Získavanie informácii z whois odpovedí

Už som popísal, že `whois` odpoveď má riadkový charakter a skupiny riadkov je niekedy možné zoskupiť do objektov. Riadkom je tým pádom možné priraďovať prioritu podľa bloku, v ktorom sa nachádzajú.

Algoritmus skúma `whois` odpoveď jednotne, nezávisle na konkrétnej regionálnej databáze. Postup je možné zhrnúť do nasledujúcich bodov:

Pre každý riadok odpovede:

1. Ak je riadok komentár, ignoruje ho
2. Ak je riadok prázdny a riadok bezprostredne za ním je neprázdny, zaznamená začiatok nového bloku.
3. Zistí, či riadok neobsahuje takú položku, ktorá obsahuje žiadanú informáciu (rozsah, krajinu...). Túto informáciu (ne)akceptuje na základe bloku, v ktorom sa nachádza.

Určenie konkrétnej informácie, ktorá bude použitá ako výstup algoritmu, je teda vykonané kontextovo, s ohľadom na umiestnenie v bloku. Presný postup sa ale líši položku od položky, preto je vhodné ich jednotlivo popísať. Najskôr však musím stručne popísať jednotlivé typy objektov.

3.3.4.1 Použité objekty

Koncept objektov nie je nikde formalizovaný, preto budem mať celý popis neformálny charakter.

Kvôli orientácii som sa rozhodol pomenovávať objekty podľa názvu prvého riadku ktorý obsahujú.

Potom som zisťoval, aké objekty sa vyskytujú v jednotlivých odpovediach (a teda databázach). Robil som to empiricky. Pýtal som sa na náhodné IP adresy a skúmal odpovede so zameraním na detekciu blokov. Pre uľahčenie práce som používal skripty, ktorých ukážku je možné nájsť v prílohe tejto práce.

Keď som spoznal objekty, musel som zistiť aké informácie mi môžu poskytnúť. Zistil som, že okrem základných informácii o adresách a vlastníkoch sa dajú získať viaceré bezpečnostné informácie slúžiace na verifikáciu zdroja, alebo zefektívnenie smerovania.

Pre mňa zaujímavé boli nasledovné objekty:

Názov	Popis
`Inetnum`	V tomto bloku sa zvyčajne popisuje rozsah, do ktorého patrí skúmaná adresa, avšak taký, ktorý bol pridelený konkrétnemu subjektu (univerzite, firme...).
`Netragne`	Popisuje väčší blok adries. Blok popísaný v `inetnum` je jednou jeho podmnožinou. Informácie v tomto bloku obsiahnuté používam iba v prípade, že `inetnum` dané informácie neposkytne.

Network	Ide o blok obsahom podobným bloku inetnum. Má iný formát zápisu a poskytuje presnejšie a automaticky ľahšie zistiteľné informácie. Je používaný hlavne ARIN databázou. Americké adresy tvoria nezanedbateľnú časť celkového priestoru, je potrebné s ním počítať. Výskyt blokov network a inetnum sa navzájom vylučuje.
Person	Blok person obsahuje údaje o správcovi adresného rozsahu, vrátane jeho mena a adresy. Správcom je osoba, ktorá má byť kontaktovaná ak dôjde k nejakej poruche, napríklad výpadku smerovania, zariadenia z príslušného dresového rozsahu. V rámci algoritmu mu neprikladám prioritu pred blokom inetnum, ale je zdrojom doplňujúcich informácii.
Organisation	V prípade, že príslušný adresný priestor bol pridelený nejakej organizácii, tento blok obsahuje údaje práve o nej. Na rozdiel od bloku person sú tu obsiahnuté kontaktné údaje na inštitúciu, nie na človeka.
Orgname	Súvisí s blokom netrange. Obsahuje informácie o organizácii spravujúcej celý veľký adresný priestor.

3.3.4.2 Rozsah

Algoritmus sa pokúša v analyzovanom riadku vyhľadať údaj o rozsahu iba v prípade, ak je aktuálny blok inetnum, network, alebo orgname. Zisťuje to v troch krokoch vykonaných sekvenčne tak, ako sú uvedené:

1. Aktuálny riadok obsahuje položku inetnum a príslušný rozsah.
2. Aktuálny blok je network a položka autharea
3. Aktuálny riadok obsahuje položku netrange a v predošlých riadkoch (blokoch) sa mu nepodarilo nájsť žiaden údaj o rozsahu.

V prvom prípade riadok obsahujúci položku inetnum súčasne znamená začiatok nového bloku, takže nemá význam kontrolovať aktuálny blok.

Hľadanie informácie v bloku netrange sa nevykoná ak už bol nejaký rozsah nájdený. Bloky netrange zvyknú byť na začiatku odpovedí a bloky intetnum, naopak, ku koncu odpovede. Ak v odpovedi chýba blok inetnum, použije sa informácia z bloku netrange. V opačnom prípade je zabezpečené, že nedôjde k prepisu konkrétnejšej informácie všeobecnejšou.

3.3.4.3 Krajina

Informácia o krajine je algoritmom hľadaná v blokoch inetnum a network, prípadne v bloku person.

Informácia o krajine nachádzajúca sa v bloku `netrange` nie je akceptovaná. Dôvodom je skutočnosť, že blok `netrange` pojednáva o správcovi celej oblasti, ktorý nemusí sídliť v rovnakej krajine ako skúmané zariadenie.

Navyše, nepripustenie tejto možnosti umožňuje zjednodušiť spôsob rozhodovania o vlastnom akceptovaní informácie:

1. Ak aktuálny riadok obsahuje položku country a aktuálny blok je `inetnum` alebo `network`, uvedenú informáciu automaticky akceptujem
2. Ak aktuálny riadok obsahuje položku country a aktuálny blok je `person/organisation/orgname`, uvedenú informáciu akceptujem, ak som doteraz žiadnu neakceptoval.

Údaje o krajinách sú reprezentované dvojznakovým kódom krajiny tak ako ich eviduje organizácie IANA. Algoritmus musí túto informáciu preložiť do pre človeka porozumiteľnejšieho formátu, to je už ale záležitosť implementácie.

3.3.4.4 Mesto

Určenie mesta na základe `whois` odpovede je najťažšou úlohou spomedzi tu spomenutých atribútov. Informácia o meste býva spravidla súčasťou adresy nejakého subjektu.

Položky, v ktorých sa môže adresa vyskytovať sú síce známe, ale formát samotnej adresy sa líši takmer subjekt od subjektu. Je možné, že každý vlastník rozsahu adries si môže sám špecifikovať aké informácie a v akom formáte budú tieto položky obsahovať. To je však iba moja špekulácia, každopádne to znamená nepríjemnosti pre každého, kto by sa rozhodol tieto údaje automaticky spracovať.

Ak sa aj podarí názov mesta určiť, neznamená to, že skúmané zariadenie sa v zistenom meste skutočne nachádza. Môže byť identifikovaná (poštová) adresa poskytovateľa internetu, ktorý sídli v zistenom meste, ale jeho klienti sú roztrúsení po okolitých dedinách. Podobne, môže ísť o celoštátneho poskytovateľa so zákazníkmi po celom štáte.

Úlohou algoritmu – vzhľadom na zameranie práce – nie je úplne presne určiť polohu, takže stačí získanie aspoň približnej informácie. Pokusy o spresnenie tejto informácie poskytujú priestor na nadviazanie na túto prácu.

Napriek vyššie uvedeným skutočnostiam som sa pokúsil navrhnúť istý spôsob pre odhad mesta. Jeho úspešnosť bude diskutovaná v jednej z nasledujúcich kapitol venovanej testovaniu môjho riešenia.

Podobne ako predošlé atribúty, aj určenie mesta záleží na aktuálne skúmanom bloku.

Blok `network` je navyše špecifický v tom, že ako jediný z použitých blokov obsahuje pomerne formalizovanú štruktúru, ktorá obsahuje položky pre každý atribút. Jednou z takýchto položiek je položka City. Dá sa ľahko identifikovať a použiť.

Ak odpoveď neobsahuje blok `network`, použijem bloky `inetnum`, person a `organisation`. Blok `inetnum` má prioritu a údaj z ostatných dvoch použijem iba v prípade, že som v predošlých riadkoch žiadnu informáciu o meste nenašiel.

V bloku `inetnum` hľadám položky `descr` a v blokoch person a `organisation` položky `address`. Ich hodnoty už analyzujem rovnako nasledujúcim spôsobom:

1. Ak je hodnota položky `descr`/`address` reťazec bez medzier a čísel a jeho dĺžka je dlhšia ako X, považujem ho za názov mesta.
2. Riadok obsahuje položku `descr`/`address` a jej hodnota je zložená z čísla a reťazca bez medzier a čísiel, tento reťazec považujem za mesto.
3. Podobne ako v predošlom kroku ale s opačným poradím čísla a reťazca medzier.

Prvý krok je zameraný na riadky, ktoré obsahujú iba názov mesta. Druhé dva počítajú s inými údajmi v okolí názvu mesta, akým je napríklad smerové číslo.

Názov mesta je reťazec. Na tento reťazec nie sú kladené žiadne požiadavky, takže podmienke dlhší ako X vyhovuje viac riadkov. Hlavne v prípadoch, že je adresa rozdelená do viacerých riadkov. Preto sú potrebné zvyšné dva kroky. Stále však existujú adresy, ktoré majú tak exotický formát, že ich tieto tri kroky nedešifrujú.

Existujú aj záznamy adries, ktoré majú iba jeden riadok, a jednotlivé položky oddelené čiarkami. Transformovať a dešifrovať takéto riadky je ďalšou z možností ako vylepšiť presnosť algoritmu. Na niektoré takéto riadky sa ale stále dajú aplikovať uvedené pravidlá a vrátiť vierohodný výsledok.

3.3.4.5 Región

Položku región som do algoritmu zaviedol, pretože som očakával, že budem schopný nejakú informáciu o ňom zistiť.

Ukázalo sa však, že to nie je také jednoduché. Použiteľný je iba blok `network`, kde sa vyskytuje položka `region` v zmysle konkrétneho amerického štátu.

Pre zistenie regiónu v zmysle akom ho chápeme v Čechách by algoritmus potreboval použiť nejaký iný zdroj informácii.

3.4 Vylepšenia algoritmu

V čase písania tejto podkapitoly som už dokončil implementáciu navrhnutého algoritmu. Od začiatku návrhu až po tento moment som narazil na niektoré vylepšenia, ktoré by mohli pomôcť k spresneniu dosiahnutých výsledkov.

3.4.1 Detailnejšie spracovanie whois

Prvá kategória sa týka `whois` databáz. Spomenul som, že existuje päť organizácii spravujúcich tieto databázy. Každá používa iný formát uloženia dát. Ak by bola navrhnutá a implementovaná verzia analýzy `whois` odpovede pre každú organizáciu samostatne, je pravdepodobné, že by bolo dosiahnutých presnejších výsledkov. Odpadla by nutnosť zamerať sa na všeobecnú analýzu, s ktorou som sa stretol v mojom prípade, čím by sa uvoľnil priestor pre hlbšie riešenie kontextového prístupu alebo samotnej identifikácie hodnôt položiek (napr. spomenuté čiarky v adresách).

3.4.2 Hlbšie využitie traceroute

Iný priestor pre vylepšenia je v použití utility `traceroute`. Okrem jednoduchého poskytnutia adries je možné detailnejšie analyzovať výstup s úmyslom zistiť, komu patria jednotlivé zariadenia a o aký typ zariadení vlastne ide. Z takýchto vedomostí sa dajú o polohách zariadení vyvodzovať rôzne závery a prispôsobovať ich dôležitosť pri určovaní výsledku.

3.4.3 Doplnenie ďalších zdrojov

Posledným spôsobom vylepšenia, nad ktorým som uvažoval, je pridanie viacerých zdrojov pre doplňujúce informácie.

Pridaním doménových mien (hostname) k skúmaným adresám by sa naskytol priestor pre ďalší spôsob určenia mesta, obzvlášť v spojení s rozšíreným využitím `traceroute`. Niektorí poskytovatelia pripojenia majú vo zvyku nazývať svoje zariadenia podľa lokalít, v ktorých sa nachádzajú. Analýzou doménového mena je teda možné určiť mesto presnejšie, než s pomocou `whois` odpovedí.

3.4.3.1 Príklad využitia hostname

Na poskytovateľa internetu, ktorý svoje zariadenia pomenováva podľa ich reálnej polohy ma upozornil pán Vrba. Ide o regionálneho poskytovateľa podnikajúceho v menších obciach v okolí Brna. Vo `whois` databáze je možné nájsť záznam:

```
inetnum:        217.197.158.0 - 217.197.158.255
netname:        ELKOMP-NET
descr:          Local ISP Kromeriz
country:        CZ
admin-c:        IS460-RIPE
tech-c:         IS460-RIPE
person:         Ivo Stanek
address:        Ivo Stanek - ONLINE
address:        Baska 467
address:        739 01 Baska
address:        Czech Republic
```

Týmto spôsobom je možné získať množinu adries patriacemu tomuto poskytovateľovi. Následne je možné vykonať DNS dotaz na každú adresu, ideálne funkciou podobnou `gethostbyaddr`, ktorá preloží IP adresu na príslušné doménové meno.

Zo zisteného rozsahu sa mi podarilo zistiť doménové meno iba 78 adresám. Z toho niektoré adresy mali priradený totožný hostname a niektoré neobsahovali potrebnú informáciu. Preto uvediem iba tie, ktoré sú relevantné pre tento príklad:

IP	Hostname	IP	Hostname
217.197.158.18	web.**korycany**.net	217.197.158.83	**prasklice**.elkomp.net
217.197.158.20	router.**strilky**.net	217.197.158.84	**nitkovice**.morkovsko.net
217.197.158.21	sw1**hradisko**.elkomp.net	217.197.158.85	**paclavice**1.elkomp.net
217.197.158.22	server.sou-**koryc**.cz	217.197.158.86	**pornice**.elkomp.net
217.197.158.28	**hradisko**.longnet.info	217.197.158.88	**litencice**1.elkomp.net
217.197.158.58	skola.**nesovice**.cz	217.197.158.91	**litencice**2.elkomp.net
217.197.158.61	skola.**brankovice**.cz	217.197.158.92	**strabenice**.elkomp.net
217.197.158.67	**bojanovice**.elkomp.net	217.197.158.93	server.zs**morkovice**.cz
217.197.158.68	**zlobice**.elkomp.net	217.197.158.94	**paclavice**2.elkomp.net
217.197.158.74	IP1.**morkovice**.net	217.197.158.82	**uhrice**.elkomp.net

Zistené záznamy teda obsahujú názvy obcí. Ľudský čitateľ ich identifikuje jednoducho a na prvý pohľad. Pre stroj táto úloha nie je jednoduchá, keďže nie vždy je informácia na rovnakom mieste (rovnakej úrovni subdomény). Občas sú tiež nejakým spôsobom skrátené alebo je k ním pridaný iný znak.

Tento poskytovateľ je však rarita. Nepodarilo sa mi identifikovať ďalších poskytovateľov, ktorý by podobnú konvenciu používali.

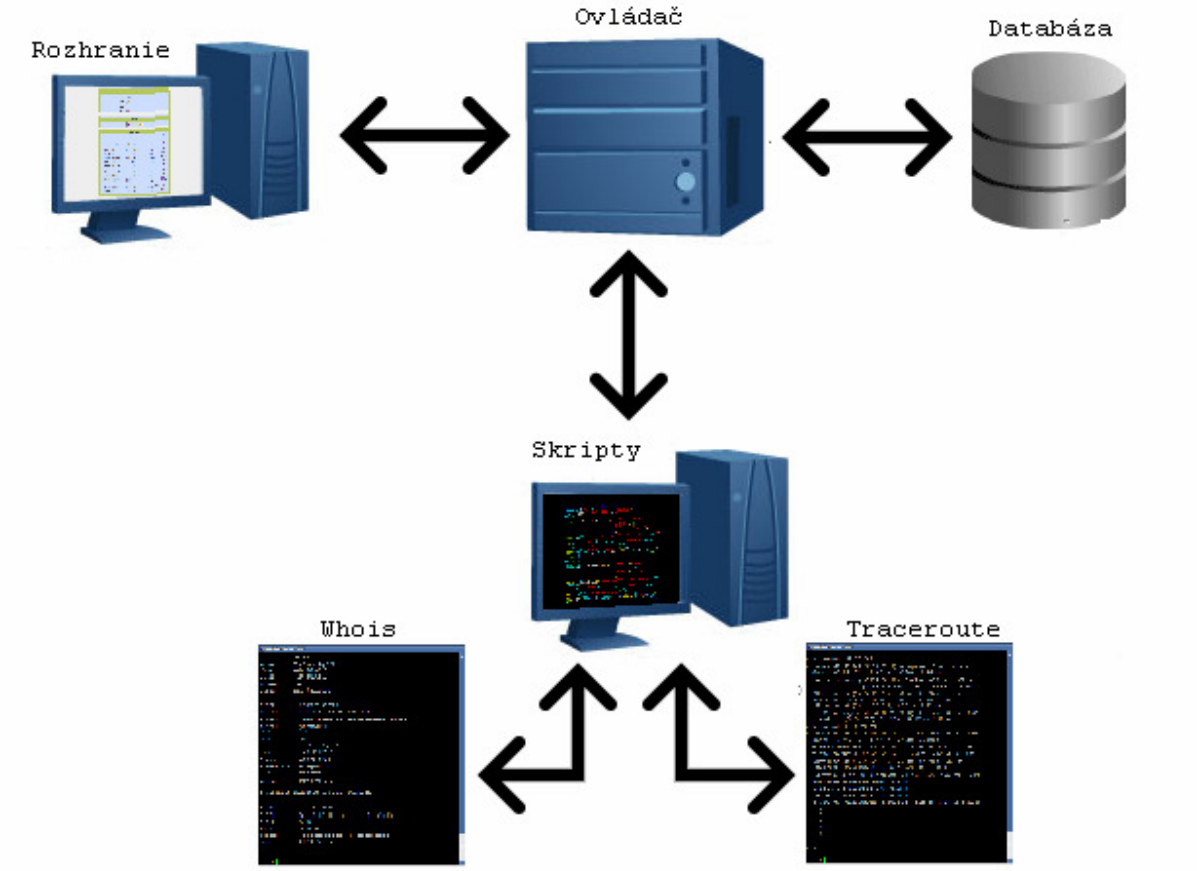
4 Implementácia

V tejto kapitole popíšem implementáciu algoritmu ako súčasti celého riešenia zahŕňajúceho rozhranie (hlavne kvôli testovacím účelom) a databázu, v ktorej sú uložené získané výsledky.

Okrem toho spomeniem aké technológie som použil a prečo.

4.1 Testovací systém

Testovací systém je možné rozdeliť do nasledujúcich komponentov:



4.1.1 Rozhranie

Hlavnou požiadavkou, ktorú som kládol na rozhranie, bolo urobiť ho čo najmenšie a najjednoduchšie, avšak so zachovaním potrebnej funkcionality. Potreboval som získať určitú vzorku adries potvrdených užívateľom.

Preto som sa rozhodol vytvoriť rozhranie ako internetovú stránku. Je zrejmé, že by mi nestačila statická stránka v podobe jednoduchého HTML dokumentu. Preto som použil skriptovací jazyk PHP, ktorý mi umožnil implementovať všetku potrebnú funkcionalitu.

S pomocou PHP som vedel od návštevníka stránky automaticky získať jeho IP adresu a posunúť ju ďalej. Okrem toho rozhranie obsahuje jednoduchý formulár, ktorý umožňuje užívateľovi zadať vlastnú adresu. Takto zadaná adresa je chápaná rovnako ako užívateľova adresa.

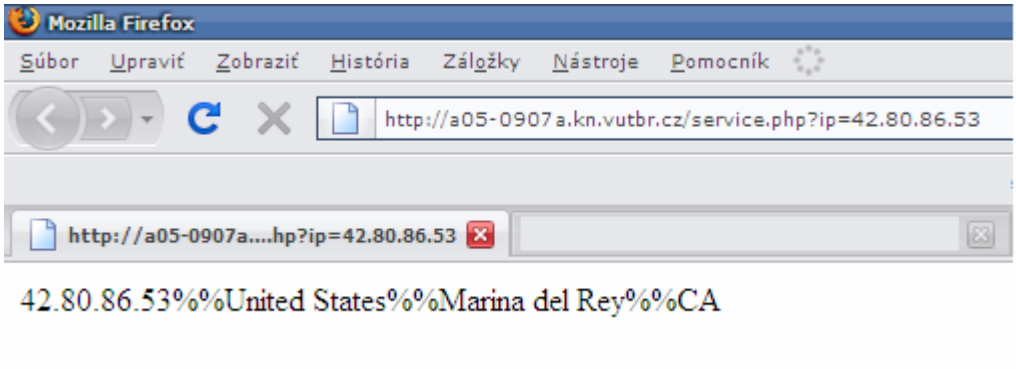
Použitím istých možností PHP sa mi občas podarilo získať o adrese viac informácii, než som potreboval. Stávalo sa to v prípade, že užívateľ používal proxy server alebo jeho počítač bol umiestnený za systémom NAT. Namiesto verejnej adresy v internete som nachádzal vnútorne, spravidla privátne, adresy, ktoré sú nepoužiteľné. Kvôli vyriešeniu tohto problému som teda musel nepoužiť časť dostupnej funkcionality.

Posledným prvkom internetovej stránky je výpis obsahu databázy v dvoch častiach. Prvá zobrazuje relatívne vierohodné poskúmané adresy, druhá adresy, ktorých analýza neprebehla vierohodne.

Obrázok 3.1: Ukážka rozhrania – internetová stránka

1. Údaje o adrese návštevníka stránky
2. Priestor pre zadanie ľubovoľnej adresy alebo adresy stránky. A výpis prípadne zistených informácii Formulár implicitne obsahuje náhodne vygenerovanú IP adresu
3. Výpis vyskúmaných adries. Používal som ho hlavne na testovacie účely.

Okrem užívateľky príjemnej varianty rozhrania som implementoval verziu, ktorá je ľahšie spracovaná automaticky. Ide o istú formu webovej služby, kde potenciálny klient iba zadá URL s príslušným parametrom a je mu vrátená odpoveď.



Obrázok 3.2: Ukážka rozhrania – webová služba

Obrázok bol síce zachytený pomocou internetového prehliadača, ale dôležité aspekty sú viditeľné. Vrátená odpoveď nie je HTML dokument ale obyčajný text. Je formátovaný tak aby ho bolo možné jednoducho spracovať a identifikovať jednotlivé položky.

Aby rozhranie vedelo získať údaje o vykonaných analýzach, potrebuje sa pripojiť k databáze. Toto pripojenie je použité aj ovládačom. V tomto mieste sa ponúka zjednotenie rozhrania a ovládača do jednej komponenty. Ich úloha je však natoľko odlišná, že som sa kvôli názornosti rozhodol ponechať ich oddelené.

4.1.2 Ovládač

Tento komponent obsahuje implementáciu samotného algoritmu. Podobne ako rozhranie, je implementovaný v jazyku PHP, čím som zjednodušil spôsob komunikácie medzi rozhraním a ovládačom.

Napriek jednoduchosti a malému rozsahu problému som sa rozhodol použiť pre tento komponent objektový prístup. Jeho aplikácia relatívne sprehľadnila zdrojový kód.

Komponent je reprezentovaný jednou triedou, nazvanou `locator`. Jej atribúty sú tvorené množinou vstupných a výstupných parametrov. Rozhranie potom jednoducho vytvorí novú inštanciu triedy `locator` (presnejšie, príslušného potomka, viď nižšie), pričom ako parameter pre konštruktor použije skúmanú adresu. Následne zavolá metódu `locate`, ktorá vykoná samotný algoritmus a nastaví hodnoty výstupných parametrov.

Implementácia ďalej obsahuje dve triedy, ktoré sú potomkami triedy `locator`. Obsahujú iba implementáciu výpisu výsledku formátovanú HTML, ktorý je odlišný pre adresu klienta a pre adresu náhodne skúmanej adresy, čo je viditeľné na ukážke rozhrania.

Zaujímavá je časť, v ktorej Ovládač pristupuje k analýze výstupu utility `traceroute`. V tomto mieste by bolo možné samotný výskum vykonávať tak, že by sa tvorila nová inštancia `lokátoru` pre každú adresu. Problematická by však mohla byť situácia kedy by každá nová inštancia znovu a znovu vykonávala `traceroute`, čo by spôsobovalo neúnosne spomalenie.

Preto som sa rozhodol `traceroute` zavolať iba raz a potom v cykle spracovávať jeho výsledok.

4.1.3 Databáza

Riadiaca časť je implementovaná pomocou PHP, tým pádom je pomerne výhodné použiť databázový systém MySQL.

Výhodou je jednoduchý spôsob integrácie do použitého jazyka. Ide o voľne dostupný software, čo eliminuje prípadné licenčné problémy. Hlavnou vlastnosťou MySQL databáz je ich jednoduchosť. V porovnaní s komerčnými produktmi síce neposkytuje takú veľkú funkcionalitu, tú však v rámci tejto práce aj tak nepotrebujem.

Výnimkou je databáza PostgeSQL, ktorá mi bola odporučená Ing. Vrbom. Obsahuje dátový typ IP adresa, ktorý by bol pre mňa užitočný, hlavne kvôli definovaným operáciám porovnania.

Z dôvodu predošlej skúsenosti s MySQL a spôsobom implementácie zvyšku systému som sa rozhodol napriek tomuto odporučeniu nevyužiť PostgreSQL. Absenciu spomenutej užitočnej vlastnosti som musel vyriešiť ináč.

Používal som dve tabuľky. Jednu pre spoľahlivo určené adresy, druhú pre adresy, ktoré algoritmus označil za nevierohodné. Ich štruktúra je však rovnaká:

Položka	Typ	Popis
`id`	`int`	primárny kľúč
`range_min`	`integer unsigned`	začiatok rozsahu, do ktorého patrí nájdená adresa
`range_max`	`integer unsigned`	koniec rozsahu, do ktorého patri nájdená adresa
`last_ip`	`integer unsigned`	naposledy skúmaná adresa z príslušného rozsahu
`last_time_checked`	`integer`	dátum vykonania poslednej analýzy adresy a teda rozsahu
`country`	`varchar(50)`	zistená krajina
`city`	`varchar(50)`	zistené mesto
`region`	`varchar(50)`	zistený región

Údaje o adresách ukladám ako jednoduché prirodzené čísla. Rozhranie a ovládač manipulujú s IP adresami vo formáte textového reťazca. Pri zisťovaní, či sa zadaná adresa patrí do rozsahu prítomného v databáze, je však nutné adresy porovnať. Operácia porovnania nad textovými reťazcami, ktorá by sa použila, v tomto prípade nepracuje korektne. Podľa väčšiny implementácie funkcií pre porovnanie reťazcov platí, že:

```
10.10.010.10 < 10.10.10.10
```

Čo je síce správne, ale ak reťazec chápem ako IP adresu, výsledok nie je správnym pretože uvedené adresy sú totožné. PHP a MySQL obsahujú zabudované funkcie slúžiace na prevody adries na celé prirodzené čísla tak aby každej adrese odpovedalo práve jedno číslo. To mi umožní porovnávať adresy ako obyčajné čísla.

4.1.4 Skripty analyzujúce whois a traceroute

Analýza výstupu utilít `whois` a `traceroute` je založená na spracovaní textu. Najvýhodnejší mne známy prostriedok pre túto úlohu je skriptovací jazyk Perl.

Teoreticky by bolo možné implementovať ten istý postup v PHP, ktoré obsahuje prostriedky nevyhnutne potrebné pre túto úlohu, dokonca podporuje syntax používanú v jazyku Perl. Ten je však oveľa pohodlnejší a disponuje bohatším aparátom pre prácu s regulárnymi výrazmi, ktoré sú v podstate synonymom pre akúkoľvek analýzu textu. Príkladom sú napríklad špeciálne premenné `$1,$2,…` , ktoré používam veľmi často.

4.1.4.1 Regulárne výrazy

Regulárne výrazy požívam vo svojej implementácii pomerne často. Pomocou reg. výrazu kontrolujem, či je vstupný reťazec korektná IP adresa. Tento RV výraz má tvar (s použitím syntaxe jazyka Perl):

```
25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?
```

Čo zodpovedá číslu od 0 do 255.

Perl umožňuje tento reťazec uložiť do premennej, premenné zreťaziť s inými premennými alebo reťazcami. Ak uvedený reťazec označím ako `$atom`, celú adresu dostanem nasledovne:

```
$atom."\.".$atom."\.".$atom."\.".$atom
```

Podobný spôsob, avšak s mierne odlišnou notáciou môžem použiť aj v PHP, kde je žiaduce, aby som chybne zadanú adresu rozpoznal čo najskôr.

Overovanie IP adries však ani zďaleka nie je jediným prípadom, kde využívam reg. výrazov. V podstate detekcia položiek tak ako bola popísaná v kapitole o návrhu algoritmu je vykonaná práve pomocou RV. Uvediem príklad niektorých položiek:

Regulárny výraz	Zisťovaná položka
`[Cc]ountry:\s+(\w+)$`	krajina
`[iI]net[nN]um:\s+($ip_range)`	rozsah
`^descr:\s+(\S+)$`	mesto ako súvislý reťazec
`\W(\S+)\s\d+`	mesto ako reťazec suediaci číslu

4.1.4.2 Cykly

Popisoval som postup spracovania `whois` odpovede, ktorý je založený na riadkoch a vyžaduje, aby som bol schopný pracovať nielen s riadkom aktuálnym, ale aj s riadkami okolo neho. Preto nemôžem použiť nečíslované cykly jazyka Perl (`foreach`), ale musím sa uchýliť k cyklom typickejším pre jazyk C (`for`), ktoré nemusia byť nutne optimálne. Implementácia našťastie nekladie dôraz na maximálnu efektívnosť, preto som si mohol takéto drobné zdržania dovoliť.

4.1.4.3 Transformácie

Niektoré `whois` odpovede neobsahovali údaj o danom rozsahu vo formáte

```
                IP adresa - IP adresa
inetnum:        147.229.0.0 - 147.229.255.255
```

Ale ako vo formáte

```
                IP adresa/X
network:IP-Network:68.201.128.0/17
```

Kde X je jedno alebo dve číslice znamenajúce počet bitov IP adresy, ktoré sú použité pre vytvorenie masky siete.

Takýto údaj je pre mňa nepoužiteľný, preto som ho musel transformovať do použiteľnej podoby. Perl mi výrazne uľahčil tento problém, pretože celý mechanizmus sa mi podarilo zapísať v niekoľkých riadkoch.

Druhou transformáciou, ktorú som potreboval urobiť, je prevod kódu krajiny na jej zrozumiteľný názov. V samostatnom súbore som si uložil dvojice kód – krajina. Na samotnom konci analýzy, tesne pred vrátením výsledku, som daný súbor načítal a zistil odpovedajúcu krajinu, tú som potom použil na výstup.

4.1.4.4 Formát výsledku a jeho prevzatie

Použité perlové skripty vypisujú výsledok jednoducho na štandardný výstup, každú položku na samostatný riadok.

Keďže sú volané z prostredia PHP funkciou `exec`, všetok ich výstup je zmenený na jeden reťazec a použitý ako návratová hodnota `exec` funkcie.

Aby som vedel s týmto výsledkom zmysluplne pracovať, potrebujem zavolať funkciu `explode`, ktorá rozdelí reťazec na časti – v tomto prípade sú časti oddelené znakom `\n` – a uložené do poľa, z ktorého ich môžem jednoducho použiť.

4.1.5 Použitý softvér a spôsob fungovania celého systému

Jedným z najväčších problémov, ktoré som v rámci tejto práce musel riešiť, bolo nájsť spôsob ako spustiť externý skript z prostredia PHP.

Väčšina komerčných a verejných hostingov, vrátane školských serverov, majú túto funkciu z bezpečnostných dôvodom zakázanú. Preto celý systém bežal na mojom súkromnom počítači, kde som mal spustené všetky nevyhnutné komponenty:

- Webový server Appache s interpretom PHP
- MySQL databáza
- Interpret jazyka Perl

Používam operačný systém MS Windows XP. Jeho súčasťou síce je implementácia utility `traceroute`, klient `whois` mi však chýbal.

Zvažoval som rôzne alternatívy. Existujú napríklad balíky pre PHP, ktoré plnia túto funkciu, alebo som mohol sám naprogramovať jednoduchého `whois` klienta. Nakoniec som sa však použiť aplikáciu `cygwin`, ktorá emuluje unixové prostredie v rámci operačného systému Windows. Jej súčasťou je aj potrebný `whois` klient.

Toto všetko komplikuje pozíciu vedúcemu mojej práce rovnako ako jeho oponentovi, ktorí pri hodnotení práce budú pravdepodobne potrebovať systém spustiť a overiť jeho funkčnosť. V prílohe tejto práce je možné nájsť kompletné zdrojové kódy spolu s podrobnými pokynmi. Alternatívne je možné na základe dohody využiť môj počítač počas hodnotenia práce.

Konečné rozhodnutie závisí na dohode s vedúcim práce.

4.2 Záverečná poznámka k implementácii

Na záver by som chcel vyjadriť svoje sympatie k jazyku Perl. Jazyk umožňuje zápis zložitých postupov niekoľkými príkazmi, ktoré sú extrémne šetrné na priestor a celkový počet riadkov. Pri tom všetkom sa však nestráca čitateľnosť kódu.

Ďalším pre mňa fascinujúcim aspektom je rozmanitosť jazyka. Jednu úlohu je možné zapísať viacerými spôsobmi a platí, že čím má autor kódu väčšiu prax, tým je napísaný kód kratší a elegantnejší.

5 Testovanie systému

Počas testovania som sa zameral na porovnávanie svojich výsledkov s výsledkami ostatných produktov. Nezaoberal som sa podrobným testovaním rýchlosti reakcie pretože na ňu nie je kladený až taký dôraz.

Vykonával som dve skupiny testov. V prvej som skúmal adresy, ktorých výsledky som mal možnosť overiť ľuďmi, ktorí dané adresy reálne používajú. Zhodou okolností som mal v období písania tejto práce viacero spolužiakov na výmenných pobytoch v zahraničí. Okrem toho som požiadal o pomoc iných známych, ktorí sú dlhodobo v zahraničí. Počet takto preskúmaných adries je však pomerne limitovaný.

Druhú množinu adries som zostavil z adries rôznych univerzít po celom svete. Keďže ide o verejné inštitúcie, ich poloha sa dá overiť, nemožno sa však spoliehať na úplnú presnosť.

5.1 Adresy potvrdené užívateľmi

5.1.1 Požité adresy

Adresa	Popis
147.229.220.177	KolejNet, VUT
86.194.88.11	Adresa z Nice, Francúzsko
193.85.232.219	IP adresa poskytovateľa, ktorého klient je v Brne
139.91.68.39	univerzitné pripojenie, Heraklion, Kréta, Grécko
131.111.203.113	Univerzitné pripojenie, Cambridge, Anglicko
213.151.245.186	poskytovateľ v Piešťanoch, Slovenská republika
90.202.33.217	komerčné pripojenie, Londýn
195.154.108.63	pripojenie na internáte, predmestie Paríža
147.215.8.26	Univerzitné pripojenie, predmestie Paríža

5.1.2 Určenie rozsahu

Služby *Ip2location* a *GeoIP* neposkytujú informácie o rozsahoch, preto uvádzam iba výsledky poskytnuté mojím riešením a *NetGeom*.

Adresa	Moje riešenie	NetGeo
147.229.220.177	147.229.0.0 - 147.229.255.255	147.229.0.0 - 147.229.255.255
86.194.88.11	**86.194.88.0 - 86.194.88.255**	**82.0.0.0 - 95.255.255.255**
193.85.232.219	**193.85.232.216 - 193.85.232.223**	**193.85.232.0 - 193.85.232.255**
139.91.68.39	139.91.0.0 - 139.91.255.255	139.91.0.0 - 139.91.255.255
131.111.203.113	131.111.0.0 - 131.111.255.255	131.111.0.0 - 131.111.255.255
213.151.245.186	**213.151.244.0 - 213.151.245.255**	**213.0.0.0 - 213.255.255.255**
90.202.33.217	**87.87.0.0 - 87.87.255.255**	**82.0.0.0 - 95.255.255.255**
195.154.108.63	**195.154.108.0 - 195.154.108.255**	**195.154.108.0 - 195.154.108.127**
147.215.8.26	147.215.0.0 - 147.215.255.255	147.215.0.0 - 147.215.255.255

Vo vyznačených riadkoch sa moje výsledky nezhodujú s *NetGeom*. Niektoré sú odlišné a červený riadok je úplne zlý. Pokúsim sa identifikovať možné príčiny:

- `86.194.88.11, 193.85.232.219, 213.151.245.186` - v čase, keď bola *NetGeo* databáza naposledy aktualizovaná, tieto adresy a ich príslušné rozsahy ešte nepatrili poskytovateľom, ktorým patria dnes. Preto sú v *NetGeo* evidované ako súčasti väčších blokov.
- `90.202.33.217` - môj systém vrátil úplne nesprávny výsledok. Dôvodom je chybný záznam vo `whois` databáze. Je síce nájdený a spracovaný korektne, ale uvedená hodnota je nesprávna.

5.1.3 Určenie krajiny

Údaj o krajine je poskytovaný všetkými porovnávanými službami:

Adresa	Moje riešenie	NetGeo	ip2location	GeoIP
147.229.220.177	Czech Republic	CZ	CZECH REPUBLIC	CZ
86.194.88.11	France	**US**	FRANCE	FR
193.85.232.219	Czech Republic	CZ	CZECH REPUBLIC	CZ
139.91.68.39	Greece	GR	GREECE	GR
131.111.203.113	Great Britain	UK	UNITED KINGDOM	GB
213.151.245.186	Slovakia	**NL**	SLOVAKIA	SK
90.202.33.217	Great Britain	US	UNITED KINGDOM	GB
195.154.108.63	France	FR	FRANC	FR
147.215.8.26	France	FR	FRANCE	FR

Zistenie krajiny je v porovnaní s ostatnými produktmi úspešné. Výnimkou sú dva prípady *NetGeo*, kde krajina nesúhlasí. Dôvodom je opäť neaktuálnosť použitej databázy.

5.1.4 Určenie mesta

Identifikácia mesta je už pomerne konkrétny údaj. Preto je možné očakávať viaceré nepresnosti.

Adresa	Moje riešenie	NetGeo	ip2location	GeoIP
147.229.220.177	Brno	BRNO	BRNO	Brno
86.194.88.11	Nice	MARINA DEL REY	MARTIGUES	Cagnes-sur-mer
193.85.232.219	Prague	PRAGUE	PRAGUE	Prague
139.91.68.39	Heraklio,	-	IRAKLEION	Iráklion
131.111.203.113	Amsterdam	CAMBRIDGE	CAMBRIDGE	Cambridge
213.151.245.186	customer	AMSTERDAM	BRATISLAVA	Moravany nad Váhom
90.202.33.217	2006091165	MARINA DEL REY	LONDON	Neston
195.154.108.63	-	PARIS	PARIS	Paris
147.215.8.26	Noisy-le-Grand	NOISY-LE-GRAND	NOISY-LE-GRAND	Noisy-le-grand

Výsledky sú veľmi rôznorodé. To podstatné sa pokúsim zhrnúť v nasledovných bodoch.

- Amsterdam je miesto sídla organizácie RIPE, ktorej patria všetky adresy nepridelené inštitúciám, poskytovateľom internetu a podobne. Táto skutočnosť je regulárne uvedená vo `whois` databáze, preto ju geolokačné systémy nachádzajú.
- *NetGeo* opäť trpí na svoju neaktuálnosť. Marina del Rey je pravdepodobne miesto sídla ARIN, takže je to rovnaká situácia ako pri Amsterdame.
- Moje riešenie vracia občas nezmyselné výsledky. Je to zapríčinené nedostatočnou analýzou `whoisu`, kde formát záznamov síce vyhovuje použitému regulárnemu výrazu, ale jeho obsah je nezmyselný.
- Moje riešenie je však jediné, ktoré správne identifikovalo adresu z Nice.
- *GeoIP* označil moju domácu adresu za adresu pochádzajúcu z obce Moravany nad Váhom. Ide o obec susediacu s mestom Piešťany, ktoré je správnym údajom. Ani jedna služba ho však nedokázala identifikovať

5.1.5 Určenie kraju

Môj systém síce hľadá údaj o kraji, robí to však jednoducho a nespoľahlivo. Tento údaj nie je ani ukladaný do databázy. Porovnávať výsledky ostatných produktov s mojím je teda bezpredmetné.

5.2 Inak overiteľné adresy

Okrem užívateľmi overiteľných adries som vykonával testy aj na skupine adries, ktorú som zostavil prevažne z adries stránok náhodne zvolených univerzít. Výsledky sa dajú pomerne ľahko overiť, pretože umiestnenie univerzít je známe, napríklad pomocou poštovej adresy uvedenej na stránke príslušnej univerzity.

Adresy použité v predošlej podkapitole som zbieral a testoval postupne, súčasne s implementáciou. Teraz skúmané adresy som testoval po dokončení implementácia jednorázovo. Preto som sa rozhodol pokúsiť sa testovanie automatizovať.

K tomu sa mi opäť hodil jazyk Perl, ktorý je k takýmto úlohám viac ako priateľský. Zdrojové texty testovacích skriptov sú súčasťou zdrojových kódov.

Automatizovať proces dotazovania sa mi však podarilo použiť iba pri mojom riešení a pri systéme *NetGeo*. Oba produkty poskytujú formu rozhrania, ktorá je jednoducho dostupná cez HTTP protokol a vrátené dokumenty sa ľahko spracovávajú. Zvyšné dva komerčné produkty takúto jednoduchú cestu neponúkajú a prípadná automatizácia si žiada oveľa väčšie úsilie.

Vynechal som testovanie rozsahu a kraju, keďže ich považujem za nezaujímavé. Rozhranie, ktoré použijem na komunikáciu so svojím riešením rozsah ani neposkytuje a identifikáciu kraju nemám implementovanú tak kvalitne, aby sa ju vôbec oplatilo testovať.

Použité adresy popíšem a dosiahnuté výsledky sa pokúsim analyzovať v nasledujúcich podkapitolách.

5.2.1 Použité adresy

Adresa	Univerzita	Mesto	Krajina
193.0.72.140	University of Warsaw	Warszawa	Polland
153.1.6.41	University of Tampere	Tampere	Finland
128.39.41.45	Gjøvik University College	Gjovik	Norway
200.3.114.53	Universidad Ricardo Palma	Lima	Peru
195.229.241.194	University Of Dubai	Dubai	United Arab Emirates
199.185.113.44	The King's University College	Edmonton	Canada
199.185.120.59	Concordia University College	Edmonton	Canada
146.230.128.52	University of KwaZulu-Natal	Dubran	South Africa
139.179.10.12	Bilkent University	Ankara	Turkey
130.102.6.192	The University of Queensland	Brisbane	Australia
130.194.11.149	Monash University	Victoria	Australia
128.103.60.28	Harward University	Cambridge, MA	United States
130.132.51.8	New Haven, CT	New Haven, CT	United States
193.205.206.17	University of Trento	Trento	Italy

5.2.2 Určenie krajiny

Adresa	Moje riešenie	NetGeo	ip2location	GeoIP
193.0.72.140	Poland	PL	POLAND	PL
153.1.6.41	Finland	FI	FINLAND	FI
128.39.41.45	Norway	NO	NORWAY	NO
200.3.114.53	**United States**	**AU**	**PERU**	**PE**

195.229.241.194	United Arab Emirates	NL	UNITED ARAB EMIRATES	AE
199.185.113.44	Canada	CA	CANADA	CA
199.185.120.59	Canada	CA	CANADA	CA
146.230.128.52	South Africa	ZA	SOUTH AFRICA	ZA
139.179.10.12	Turkey	TR	TURKEY	TR
130.102.6.192	Australia	AU	AUSTRALIA	AU
130.194.11.149	Australia	AU	AUSTRALIA	AU
128.103.60.28	United States	US	UNITED STATES	US
130.132.51.8	United States	US	UNITED STATES	US
193.205.206.17	Italy	IT	ITALY	IT

Proces určenia krajiny priniesol koherentné výsledky. Jedinou výnimkou je chyba môjho riešenia pri určovaní krajiny z južnej Ameriky. Je to zapríčinené formátom odpovede juhoamerického `whois` serveru, na ktorého presnú syntax som sa nezameral.

5.2.3 Určenie mesta

Adresa	Moje riešenie	NetGeo	ip2location	GeoIP
193.0.72.140	Wolczanska	WARSAW	WARSAW	Warsaw
153.1.6.41	Finland	TAMPERE	TAMPERE	Tampere
128.39.41.45	Amsterdam	-	GJOVIK	Gjøvik
200.3.114.53	Phoenix	-	**LIMA**	Lima
195.229.241.194	Box	AMSTERDAM	DUBAI	Dubai
199.185.113.44	Edmonton	EDMONTON	EDMONTON	Edmonton
199.185.120.59	Calgary	EDMONTON	EDMONTON	Edmonton
146.230.128.52	4001	DURBAN	DURBAN	Durban
139.179.10.12	Ankara	ANKARA	ANKARA	Ankara
130.102.6.192	Milton	-	BRISBANE	Brisbane
130.194.11.149	Milton	CLAYTON	MELBOURNE	Monash
128.103.60.28	Cambridge	CAMBRIDGE	CAMBRIDGE	Cambridge
130.132.51.8	New Haven	NEW HAVEN	NEW HAVEN	New Haven
193.205.206.17	Tizii,	TRENTO	TRENTO	Trento

Musím skonštatovať, že počet správnych určení mesta mojím systémom je nízky. Spôsob, ktorým identifikujem názov, nie je dostatočne sofistikovaný a žiada si vylepšenie.

Všetky tri porovnávané produkty ma prekonali. V súbore adries popísanej v predchádzajúcej podkapitole som zaznamenal podstatne väčšiu úspešnosť. Dôvodom je skutočnosť, že s adresami som pracoval pri vývoji a prispôsobil som im konkrétne výrazy, ktoré hľadám vo `whois` databázach.

6 Záver

V tejto práci som sa zaoberal problémom geolokácie.

Ako prvé som sa zoznámil s podstatou problému. Zistil som, že nejde o exaktný proces, čo je zapríčinené hlavne podstatou internetu, ktorá nekladie na geografickú polohu dôraz. Stretol som sa s dvoma formami geolokácie – snahou o presné určenie geografických súradníc a pokusom popísať polohu pomocou určenia krajiny, regiónu a mesta, o ktorou som sa zaoberal v tejto práci.

Preskúmal som produkty, ktoré sú už vytvorené a geolokáciou sa zaoberajú (ne)komerčne. Komerčné produkty skrývajú svoje metódy a ich štúdium je takmer nemožné. Nekomerčné produkty sú buď neudržované alebo neposkytujú kvalitatívne porovnateľné výsledky.

Podarilo sa mi však získať viac či menej podrobný popis postupov používaných niektorých produktov. Tiež sa mi podarilo odhadnúť spôsob akým pracujú komerčné produkty.

Všetky nadobudnuté informácie ovplyvnili spôsob, ktorým som navrhol vlastné riešenie. Účelom bolo vyskúšať si samotný návrh systému, nie presne určiť polohu. Samotné určenie polohy je v mojom návrhu založené na analýze `whois` odpovede. Tá je prispôsobená na európske a americké servery. Pridanie podpory pre ostatné regióny by si vyžadovalo ďalšiu prácu, to už však nebolo mojou úlohou.

Návrh som implementoval vrátane dvoch variantov rozhrania – užívateľsky priateľskej internetovej stránky a strojovo orientovanej webovej služby. Kvôli špeciálnym požiadavkám na webový server som musel celý prevádzkovať na svojom súkromnom počítači.

Testy ukázali, že sa mi podarilo navrhnúť systém, ktorý dokáže pomerne presne určiť krajinu, ale s určením mesta už má problém. Domnievam sa, že ide o problém nízkeho počtu vstupných dát a ich nedostatočne sofistikovaným spracovávaním. Na túto funkcionalitu však v rámci práce nebol kladený dôraz.

Počas riešenia práce som identifikoval možnosti rozšírenia, ktoré by zrýchlili odozvu, spresnili výstupy a všeobecne prispeli k spoľahlivosti systému.

Literatura

[1] WWW stránky. Systems and methods for determining collecting and using geographic locations of internet users. United States Patent 6757740. http://www.freepatentsonline.com/6757740.html [Online, dostupné 12.5.2009]

[2] WWW stránky. Geolocation by IP Address, Andrew Turner, 2004. http://www.linuxjournal.com/article/7856. [Online, dostupné 12.5.2009]

[3] WWW stránky. How to geographically locate an IP Address. http://ipinfo.info/html/geolocation.php. [Online, dostupné 12.5.2009]

[4] WWW stránky. Geolocation. From Wikipedia, the free encyclopedia. http://en.wikipedia.org/wiki/Geolocation. [Online, dostupné 12.5.2009]

[5] WWW stánky. NetGeo - The Internet Geographic Databáze. http://www.caida.org/tools/utilities/netgeo/. [Online, dostupné 12.5.2009]

[6] WWW stránky. Ip2location™. http://www.ip2location.com/. [Online, dostupné 12.5.2009]

[7] WWW stránky. Geobytes Home Page. http://www.geobytes.com/. [Online, dostupné 12.5.2009]

[8] WWW stránky. Geolocation and Online Fraud Prevention from MaxMind®. [Online, dostupné 12.5.2009]

[9] Gargano J., Weiss K.. *Whois and Network Information Lookup Service*. RFC 1834 (INFORMATIONAL), August 1995. Dostupné na http://tools.ietf.org/html/rfc1834. [Online, dostupné 12.5.2009]

[10] Randal A., Sugalski D., Totsch L.. *Perl 6 and Partit Essentials*. O'Reilly Media Inc., 2004. ISBN 0-596-00737-X

[11] WWW stánky. PHP: Manual Quick Reference PHP online. Dostupné na http://www.php.net/quickref.php. [Online, dostupné 13.5.2009]

[12] WWW stránky. Perl version 5.10.0 documentation.Dostupné na http://perldoc.perl.org. [Online, dostupné 13.5.2009]

[13] IANA. RFC 3330 : *Special-Use IPv4 Addresses.* Informational. September 2002. Dostupné na http://www.faqs.org/rfcs/rfc3330.html. [Online, dostupné 12.5.2009]

Zoznam príloh

Príloha 1. CD so zdrojovými súbormi

Obsah CD

```
\src\                   zdrojové súbory
\src\databaza\          použite tabuľky a vzorka dát
\src\prilohy\           ukážky pomocných skriptov
\src\www\               hlavná stránka, pomocne funkcie
\src\www\classes\       triedy rozhrania
\src\www\images\        definícia vzhľadu
\src\www\inc\           konfiguračné súbory
\src\www\perl\          externe skripty
\text\                  text samotnej prace
\ukazky\                ukážky prace
\readme.txt
```

Poznámka k testovaniu systému

Na spustenie systému a jeho odskúšanie je potrebné mať k dispozícii veľa softvéru (Appache, interpret jazykov PHP a Perl, MySQL databáza, klient služby whois, utilita traceroute – varianta prítomná v OS MS Windows). Navyše, je potrebné zmeniť nastavenie na nie implicitné hodnoty (spúšťanie externých skritpov v PHP, maximálny povolený čas vykonávania scriptu).
Doktor Herout ma upozornil, že to môže byť nepohodlné a nežiaduce. Keďže mám všetko potrebné dostupné na svojom počítači, dovolím si navrhnúť nasledovné:

- Ak to bude pri hodnotení práce potrebné, prídem osobne demonštrovať funkčnosť systému
- Po dohode, v určitú dobu spustím a povolím systém na svojom počítači
- Okrem toho, zložka ukážky obsahuje niekoľko uložených stránok, vygenerovaných mojím systémom. Tie môžu slúžiť ako dôkaz funkčnosti.