Západočeská univerzita v Plzni

Fakulta aplikovaných věd

Katedra geomatiky

# Diplomová práce

Vizualizace prostorových Big Data

Plzeň, 2017 Bc. Jáchym Kellar

ZÁPADOČESKÁ UNIVERZITA V PLZNI
**Fakulta aplikovaných věd**
Akademický rok: **2016/2017**

# ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

| | |
|---|---|
| Jméno a příjmení: | **Bc. Jáchym KELLAR** |
| Osobní číslo: | **A15N0006P** |
| Studijní program: | **N3602 Geomatika** |
| Studijní obor: | **Geomatika** |
| Název tématu: | **Vizualizace prostorových Big Data** |
| Zadávající katedra: | **Katedra geomatiky** |

Zásady pro vypracování:

1. Vlastnosti a definice prostorových Big Data
2. Rešerše možností vizualizace prostorových Big Data
3. Zhodnocení jednotlivých přístupů vizualizace
4. Navržení vhodné formy vizualizace datové sady SPOI
5. Tvorba ukázkové aplikace
6. Popis a analýza aplikace

Rozsah grafických prací: **dle potřeby**

Rozsah kvalifikační práce: **cca 45 stran**

Forma zpracování diplomové práce: **tištěná**

Seznam odborné literatury:


- **ELDAW, Ahmed, MOKBEL, F. Mohamed (2015). The Era of Big Spatial Data: A Survey. DBS Journal. The Database Society of Japan, 13(1), s. 25-36.**
- **LIU, Zhicheng, JIANG, Biye, HEER Jeffrey (2013). imMens: Real-time Visual Querying of Big Data. In: Computer Graphics Forum. Wiley Online Library, 32(3), s. 421-430.**


Vedoucí diplomové práce: **Ing. Mgr. Otakar Čerba, Ph.D.**
Katedra geomatiky

Datum zadání diplomové práce: **3. října 2016**

Termín odevzdání diplomové práce: **19. května 2017**

Doc. RNDr. Miroslav Lávička, Ph.D.
děkan

L.S.

Západočeská univerzita v Plzni
Fakulta aplikovaných věd

Doc. Ing. Václav Čada, CSc.
vedoucí katedry

V Plzni dne 3. října 2016

# Prohlášení

Tímto předkládám k posouzení a následné obhajobě diplomovou práci vypracovanou na závěr navazujícího studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni. Prohlašuji, že jsem diplomovou práci vypracoval samostatně pod odborným dohledem vedoucího diplomové práce a výhradně s využitím uvedené literatury a dalších informačních zdrojů.

V Plzni dne 9. května 2017 .....

podpis

## Poděkování

Tímto bych rád poděkoval vedoucímu práce Ing. et Mgr. Otakaru Čerbovi, Ph.D. za odborné vedení práce, cenné připomínky, nápady a podněty. Poděkování patří i mým rodičům za jejich podporu po celé mé studium.

## Abstrakt

Diplomová práce se zabývá možnostmi vizualizace větších objemů prostorových dat a prostorových Big Data. Představuje a srovnává nejen konkrétní nástroje z oblasti distribuovaných systémů a cloudových služeb, ale i nástroje umožňující vizualizovat objemnější prostorová data na jediném počítači. V praktické části je poté vytvořena webová aplikace prezentující datovou sadu Smart Points of Interest. Využito je přitom převodu dat do formy vektorových mapových dlaždic, která jsou následně v prostředí internetu distribuována přes mapový server a vykreslena na straně klienta pomocí WebGL. Tento přístup umožňuje plynulý interaktivní průzkum téměř 29 milionů bodů včetně zobrazení všech jejich popisných atributů.



## Klíčová slova

vizualizace, prostorová Big Data, vektorové dlaždice, Smart Points of Interest, Tippecanoe



## Abstract

This diploma thesis is concerned with the possibilities of visualization of large volumes of spatial data and geospatial Big Data. It presents and compares not only the specific tools within the area of distributed systems and cloud services, but also the tools allowing to visualize larger spatial data on a single computer. The practical part of this thesis is focused on creating web application presenting Smart Points of Interest data set. For this purpose data were converted into a form of the vector map tiles, which are distributed over the Internet through tile server and are rendered on the client side via WebGL. This approach allows users an interactive exploration of nearly 29 million points including all their attributes with minimal latency.



## Key Words

visualization, geospatial Big Data, vector tiles, Smart Points of Interest, Tippecanoe

# Obsah

# Seznam použitých zkratek

API - Application Programming Interface

BSD - Berkeley Software Distribution

CESNET - Czech Education and Scientific NETwork

CPU - Central Processing Unit

CSS - Cascading Style Sheets

CSV - Comma-separated values

DOM - Document Object Model

DVD - Digital Versatile Disc

EC2 - Elastic Compute Cloud

EMR - Elastic MapReduce

EU - Evropská unie

FOAF - Friend of a Friend

FTP - File Transfer Protocol

GB - Gigabyte

GIF - Graphics Interchange Format

BSD - Big Spatial Data

ESRI - Environmental Systems Research Institute

GIS - Geographic Information System

GLSL - OpenGL Shading Language

GML - Geography Markup Language

GNSS - Global Navigation Satellite System

GPS - Global Positioning System

GPU - Graphics Processing Unit

HDF - Hierarchical Data Format

HDFS - Hadoop Distributed File System

HPCC - High Performance Computing Cluster

HQL - Hive Query Language

HTML - HyperText Markup Language

HTTP - Hypertext Transfer Protocol

IBM - International Business Machines Corporation

IEEE - Institute of Electrical and Electronics Engineers

ISC - Internet Systems Consortium

JPEG - Joint Photographic Experts Group

JRE - Java Runtime Environment

JSON - JavaScript Object Notation

KML - Keyhole Markup Language

LIDAR - Light Detection And Ranging

MB - Megabyte

MIT - Massachusetts Institute of Technology

NASA - National Aeronautics and Space Administration

OGC - Open Geospatial Consortium

OSM - OpenStreetMap

OWL - Web Ontology Language

PBF - Protocolbuffer Binary Format

PHP - PHP: Hypertext Preprocessor

PNG - Portable Network Graphics

RDF - Resource Description Framework

SPARQL - SPARQL Protocol and RDF Query Language

SPOI - Smart Points of Interest

S3 - Simple Storage Service

SQL - Structured Query Language

SSH - Secure Shell

SVG - Scalable Vector Graphics

URL - Uniform Resource Locator

UDF - User-defined Function

VGI - Volunteered Geographic Information

WebGL - Web Graphics Library

WMS - Web Map Service

XML - Extensible Markup Language

YARN - Yet Another Resource Negotiator

ZČU - Západočeská univerzita

# Seznam obrázků

# Seznam tabulek

# 1. Úvod

Pojem Big Data se v poslední době stává čím dál tím častěji skloňovaným fenoménem, který plní programy konferencí[1] a je předmětem mnoha diskusí i odborných článků[2]. Obecně můžeme říci, že jde o taková data, se kterými, zvláště kvůli jejich objemu, neumíme běžnými nástroji a v rozumném čase pracovat (Snijders et al. 2012). Důvodem vzniku Big Data je především rozvoj nových technologií a služeb, která, často i s pomocí svých uživatelů, generují v krátkém časovém horizontu obrovská množství dat. Ta mohou navíc obsahovat i prostorovou informaci a lze je poté označit jako prostorová Big Data[3]. Příkladem zdroje takových dat mohou být například letadla informující každých několik sekund o své poloze, družice posílající na Zem satelitní snímky, chytré mobilní telefony a jejich aplikace pracující téměř neustále s informacemi o svém umístění v prostoru nebo čím dál tím více populárnější sociální sítě, kam uživatelé denně přidávají miliony příspěvků, přičemž k části z nich připojují i lokalizační údaje. Pro představu: Boeing 787 generuje průměrně 500 GB dat za jeden jediný let (Shah 2014), Hubbleův vesmírný dalekohled posílá na Zem přibližně 140 GB dat týdně[4] a služba Twitter zase produkuje 500 miliony tweetů za den[5], přičemž část z nich nese i již zmíněnou prostorovou informaci. Všechna tato data pomáhají organizacím a výrobcům lépe přizpůsobit své výrobky a služby zákazníkům, optimalizovat provoz a například v případě letů zajistit i vyšší bezpečnost. Jednoduše řečeno mají tato data vysoký potenciál. Aby však nad nimi bylo možné provádět analýzy, a získávat z nich tak potřebné a nové informace, je nejprve nutné tato data efektivně uložit a účelně zpracovat. A zde nastává jeden z hlavních problémů. Vše od mobilních aplikací až po sofistikované analytické nástroje totiž potřebuje efektivní a především rychlý přístup k libovolné podmnožině dat, což je vzhledem k vlastnostem Big Data velice náročné. Ta tak představují jeden z největších problémů, ale i výzev informatiky a internetu současnosti. Například v roce 2013 patřila Big Data

[1] Například The Rise of Big Spatial Data (Symposium GIS Ostrava, 16.-18.3.2016. Technická univerzita Ostrava) nebo BSD 2016 (1st IEEE International Workshop on Big Spatial Data. IEEE International Conference on Big Data, 5.-8.12.2016. Washington D.C., USA)

[2] viz literatura a odborné články zmíněné v kapitole č. 2

[3] Kromě pojmu prostorová Big Data, z anglického Big Spatial Data (popř. Spatial Big Data), například v (Evans et al. 2014) nebo (Eldawy a Mokbel 2015), se lze setkat i s pojmem geoprostorová Big Data, z anglického Geospatial Big Data, (Lee a Kang 2015) nebo (Li et al. 2016).

[4] http://hubblesite.org/the_telescope/hubble_essentials/quick_facts.php

[5] http://www.internetlivestats.com/twitter-statistics/#trend

mezi “TOP 13” trendů IEEE Computer Society (Černý 2013) a v dnešní době její význam i kvůli neustálému globálnímu navyšování objemu dat stále roste.

Jednou z možných cest, jak Big Data a především prostorová Big Data prezentovat a získávat z nich informace, je jejich vizualizace. Ta je pro člověka nejen názornější než textový soubor se souřadnicemi, ale slouží i jako prostředek k poznávání a pochopení dosud neznámých skutečností. Příkladem může být posouzení kvality a distribuce dané datové sady a jejích podmnožin. Často totiž nepotřebujeme znát detaily, ale spíše celkový koncept a vztahy a souvislosti mezi informacemi. Aby však člověk mohl tyto poznatky získat, je nejprve nutné za pomoci speciálních nástrojů dostupná data především zpracovat a následně samotnou vizualizaci vůbec vytvořit.

Jedním z cílů této diplomové práce je tak představit a zhodnotit jednotlivé možnosti a přístupy vizualizace prostorových Big Data. Tedy jak vůbec k tomuto problému přistupovat a jaké jsou možnosti jeho řešení. Druhým cílem je poté na základě předešlé rešerše navrhnout vhodnou formu vizualizace datové sady “Smart Points of Interest” (dále jen SPOI). Ta je vytvářena především na Katedře geomatiky na Západočeské univerzitě v Plzni v rámci projektu SDI4Apps - Uptake of Open Geographic Information Through Innovative Services Based on Linked Data[6] a bude dále vyvíjena a spravována pod projektem Peregrinus Silva Bohemica - Multimediální a digitální turistický průvodce pro přeshraniční historické cesty v Bavorském lese a na Šumavě[7]. Vzhledem k objemu této datové sady dosahujícího desítek gigabajtů a obsahu velkého počtu bodů s ní však nelze pracovat a vizualizovat ji běžnými nástroji. Cílem práce je tedy na základě předešlé rešerše zvolit vhodnou formu její vizualizace a tu následně demonstrovat praktickou tvorbou ukázkové aplikace, která bude zmíněná data prezentovat.

Diplomová práce tak volně navazuje na předchozí autorovy aktivity. Ten se ve své bakalářské práci zabýval integrací dat z různých zdrojů a jejich následnou vizualizací v podobě cykloturistické aplikace (Kellar 2015). Dále se podílel na testování javascriptové knihovny WebGLayer, která slouží pro pokročilou vizualizaci geografických dat (v řádu statisíců prvků) ve webovém prohlížeči za pomoci WebGL (Ježek et al. 2017). A v neposlední řadě je také zapojen do projektů SDI4Apps i Peregrinus Silva Bohemica, včetně tvorby samotné datové sady SPOI.

[6] http://sdi4apps.eu/
[7] https://kgm.zcu.cz/aktualni-projekty/peregrinus/

Na tomto místě je také nutné objasnit používání termínu Big Data. Přímý překlad tohoto pojmu z angličtiny do češtiny je "velká data" a tento ekvivalent navrhuje i Terminologický slovník zeměměřictví a katastru nemovitostí[8]. Setkat se však můžeme i s překladem "veledata", například v (Mayer-Schönberger a Cukier 2014). Nicméně ve většině případů se stejně jako mnoho jiných pojmů z oblasti informačních technologií ani pojem Big Data nepřekládá, a proto bude i v této práci používán originální anglický název.

Text diplomové práce je strukturován do několika částí. Na první úvodní kapitolu navazuje teoretická část zabývající se problematikou prostorových Big Data. Tedy jak lze zmíněný pojem vůbec definovat, jaká mají tato data vlastnosti a které datové sady je možné označit jako Big Data. Třetí kapitola se poté věnuje jednotlivým krokům v procesu vizualizace a shrnuje nejen její přínosy, ale především také problémy, které je nutné ve spojitosti s Big Data řešit. Čtvrtá část následně nabízí různé možnosti, jak prostorová Big Data vizualizovat. Tedy jak k tomuto problému vůbec přistupovat, jak ho řešit a především jaké konkrétní nástroje lze využít. Součástí je i základní popis těchto nástrojů včetně jejich porovnání a shrnutí výhod, nevýhod či vhodnosti a náročnosti použití. Následuje praktická část této práce zabývající se vizualizací datové sady SPOI. Na základě její analýzy a předešlé rešerše jsou vybrány nejvhodnější nástroje, pomocí kterých je následně vytvořena, popsána a zhodnocena jak vizualizace statická, tak i ve formě interaktivní webové aplikace. Závěrem je diplomová práce celkově shrnuta, popsány poznatky zjištěné v průběhu její tvorby a jsou navrženy další možné kroky, kterým se lze s ohledem na vizualizaci prostorových Big Data věnovat.

Vzhledem k velkému množství softwaru, nástrojů, programových balíčků a knihoven zmíněných i několikrát v různých částech této práce, nejsou odkazy na jejich oficiální stránky či jejich samotný kód z důvodu přehlednosti součástí textu, ale součástí příloh.

[8] http://www.vugtk.cz/slovnik

# 2. Vlastnosti a definice prostorových Big Data

Jako prostorová Big Data lze jednoduše označit prostorová data, která splňují vlastnosti Big Data (Evans et al. 2014). Je tedy nutné zaměřit se nejprve na charakteristiky a definici samotných Big Data a až poté určit specifika a problémy, které přináší prostorová, případně časoprostorová složka.

## 2.1 Pojem Big Data a jeho vymezení

Ačkoliv výraz Big Data není zdaleka nový, stále neexistuje jeho žádná jednoznačná a všemi přijímaná definice. Již tento fakt sám o sobě naznačuje celkovou nejednotnost tohoto konceptu, a stejně tak i nejednoznačnost ve vymezení toho, která data lze považovat za Big Data, a která ne. V roce 2001 definoval analytik firmy META Group (dnes známé jako Gartner) Doug Laney problém Big Data jako třídimenzionální model zahrnující jejich objem, rychlost, s jakou je potřeba je zpracovat, a jejich různorodost (Laney 2011). Tyto tři základní vlastnosti bývají podle počátečních písmen anglických ekvivalentů (volume, velocity, variety) označovány jako „3V“ model charakterizující Big Data.

Základní vlastnost, tedy objem dat, říká, že je dat příliš mnoho, než abychom je mohli zpracovávat a spravovat běžnými nástroji. Tato vlastnost tedy představuje to, co si pod pojmem Big Data nejčastěji představuje široká veřejnost. Zde je však nutné zmínit, že charakteristika objemu je velice subjektivní. Hranice mezi Big Data a normálním množstvím dat se neustále pohybuje a nelze ji přesně určit (Buyya et al. 2016). Jedním z důvodů je například čas. Neustále totiž roste nejen objem běžných úložišť, ale i výkon a možnosti hardwaru a softwaru. Navíc jsou zaváděny nové technologie a nástroje, které umožňují pracovat bez problémů se stále větším objemem dat. Dalšími faktory ovlivňující hranici mezi Big Data a normálním množstvím dat jsou poté schopnosti jednotlivých uživatelů nebo společností a samotné nástroje, které mají k dispozici. Jinými slovy je tato hranice vázána na situaci, kdy technická infrastruktura dané společnosti není schopna držet krok s jejími datovými potřebami. Pro jednu společnost tak může být objem dat blížící se desítkám gigabajtů problém, zatímco pro jiné se stane velikost dat významným aspektem, který je potřeba řešit, až při objemu desítek či stovek terabajtů (Magoulas a Lorica 2009). Pojem Big Data je tedy z pohledu velikosti dat velice relativní (Buyya et al. 2016), neboť nelze přesně definovat, jak velká data musí

být, aby bylo možné je za Big Data označit. Důležitý je zde i úhel pohledu. Pro člověka, který musí analýzu dat provádět sám jako jednotlivec, tak bude význam Big Data jiný než pro firmu z oblasti informačních technologií, která disponuje výkonnými nástroji a ve svých datových skladech běžně pracuje s obrovským množstvím dat. Právě kvůli své relativnosti z pohledu objemu dat je samotný pojem Big Data často kritizován jako špatné a nepřesné označení. Více než na velikosti dat totiž záleží na jejich složitosti a dalších vlastnostech (Zikopoulos et al. 2012). Mnoho datových sad, která jsou označena jako Big Data, nepotřebuje tolik fyzického prostoru, ale jsou složité povahy, tudíž je těžké je zpracovat. A naopak s daty zabírající velký objem místa lze někdy jednoduše manipulovat (například kvůli jejich jednoduché a pevné struktuře) a nemusí být tedy za Big Data považována.

Další vlastností Big Data související s jejich velikostí je také rychlost jejich nárůstu. Množství dat v datové sadě nebo kolekci typicky narůstá v čase velmi rychle, často až exponenciálně (Holubová et al. 2015). Zároveň však v některých případech není potřeba žádná data mazat, pouze přidávat nová, z čehož vyplývá požadavek na případnou rozšiřitelnost úložného prostoru, kde je daná datová sada uložena. Problémem však může být nutnost zvýšení kapacity prostoru přímo za chodu a teoreticky až do nekonečna.

Druhou vlastností ze základního „3V“ modelu reprezentujícího Big Data je rychlost, která představuje požadavek na co nejkratší dobu jejich zpracování. V této souvislosti je dobré zmínit speciální příklad Big Data, takzvané datové proudy. Jde o nekončící rychlý tok dat, který často není možné opakovaně procházet. Je sice možné dočasně uložit určitou skupinu dat, nicméně vzhledem k jejich neustálému nárůstu není možné uložit a zpracovat je jako celek a je tedy nutné s nimi pracovat v reálném čase (Holubová et al. 2015). Požadavek na rychlost, s jakou je potřeba data zpracovat, je zde tak naprosto zásadní. Datové proudy jsou sice extrémním případem, nicméně je nutné si uvědomit, že úlohy vyžadující okamžité zpracování velkého objemu průběžně vznikajících dat jsou v oblasti Big Data mnohem běžnější než úlohy, u kterých jsou data uložena a skladována až k pozdější analýze. Typickým příkladem, kde je nutné rychle zpracovat velké množství dat v reálném čase, je hledání nebezpečných osob na letišti prostřednictvím kamer. Analýza obrazového záznamu a identifikace hledaných osob trvající hodiny by zde představovala vážné bezpečností riziko, a proto jsou na rychlost odezvy mezi přijetím a zpřístupněním zpracovaných dat k dalšímu využití kladeny

značně vysoké nároky.

Třetí základní vlastností je poté různorodost. Ta popisuje heterogenitu dat s ohledem na jejich typ, reprezentaci a sémantickou interpretaci (Akhgar et al. 2015). Nejdůležitějším aspektem, kterým je typ dat, se zde rozumí jejich rozdělení na data strukturovaná, nestrukturovaná a semistrukturovaná. Strukturovaná data mají jasný model a popis (tedy pevnou strukturu), lze je dobře ukládat, zpracovávat a analyzovat. Typickým příkladem jsou relační databáze, kde jsou data uložena v přesně definovaných a popsaných datových polích. Nestrukturovaná data naopak nemají jasně daný řád ani nejsou organizovány předem definovaným způsobem. Jejich zpracování je tak mnohdy velmi obtížné. Spadají sem například multimediální data, jako je audio, video nebo obrázky. Třetí typ, semistrukturovaná data, lze poté chápat jako průnik dvou výše uvedených. Nejsou sice spojeny s žádným přesným datovým modelem, ale přesto obsahují značky nebo jiné oddělující znaky, které představují určitý sémantický význam, případně hierarchii, textu nebo záznamu, kterého se týkají (Akhgar et al. 2015). Z tohoto důvodu jsou často označovány jako samopopisující struktury (Akhgar et al. 2015). Příkladem takových dat mohou být dokumenty ve formátech XML nebo JSON, které se často využívají jako médium pro výměnu informací mezi aplikacemi, dále logy ze zařízení nebo email, který ukázkově kombinuje data strukturovaná (přesně definovaná hlavička - odesílatel, adresát, datum apod.) i nestrukturovaná (textový obsah zprávy a přílohy).

Z výše uvedených datových typů se pojem Big Data vztahuje především k datům nestrukturovaným a semistrukturovaným. Je nutné si uvědomit, že dat nestrukturovaných je naprostá většina, podle některých odhadů tvoří 80 až 90 % všech světových dat (Hassanien 2015). Ve své surové podobě však často nejsou nijak zvlášť užitečná. Cílem je tedy získat z nich informace nebo je zpracovat pro další použití. Tento úkol je však často velice náročný. Příkladem může být problém, jak vyhledávat v databázích multimediálních dat. Klasickou metodu vyhledávání pomocí metadat nebo textových popisků zde totiž nelze aplikovat, a musí se tak zvolit naprosto jiný přístup, například porovnávání se vzory ve znalostní databázi, pokud chceme identifikovat určitou osobu v kamerovém záznamu.

Tři výše zmíněné základní charakteristiky však nejsou pro popis Big Data definitivní. Například firma IBM přišla s rozšířením původního „3V“ modelu na „4V“ přidáním vlastnosti reprezentující věrohodnost dat (anglicky “veracity”), (Claverie-Berge 2012). V klasických relačních databázových systémech je totiž věnována velká pozornost

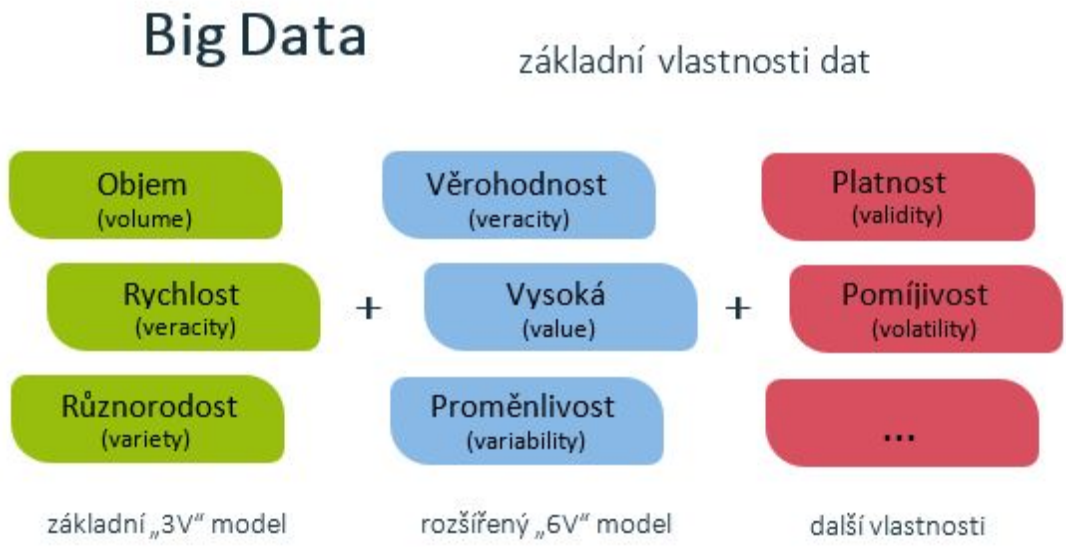
předzpracování, čištění a filtrování dat ještě předtím, než vstoupí do samotného systému. Ačkoliv tento proces není vždy bezchybný, lze výsledná data většinou považovat za konzistentní, úplná a přesná (Holubová et al. 2015). S rostoucím objemem dat a shromažďováním více informací však roste pravděpodobnost výskytu chyb a případných nejasností. Když si navíc uvědomíme, že pro oblast Big Data je typické zpracování velkého množství dat z různých zdrojů a často navíc v reálném čase, je poté jasné, že na jejich důkladné čištění a kontrolu není dostatek prostoru. Naopak je někdy tento proces i nežádoucí, jelikož některé systémy dopředu neznají všechny požadavky na využití dat a požadují tedy jejich ukládání v takové formě, ve které přicházejí (tj. "surová data"). Jakékoliv předzpracování a filtrování by poté v tomto případě snižovalo jejich hodnotu (Holubová et al. 2015). Pro následné dosažení požadované úrovně věrohodnosti je však často nutné využít robustní optimalizační techniky a přístupy, které jsou ve většině případů velmi složité a náročné. U Big Data je tedy nutné vzít v úvahu fakt, že mohou být potenciálně nepřesné nebo nekvalitní. Nemusí se sice vždy jednat o nevýhodu, ale je nutné si odpovědět na otázku, zda jim mohou společnosti a organizace věřit a dělat na jejich základě rozhodnutí. V důsledku toho se objevují i názory, zda má smysl vůbec Big Data zpracovávat, popř. zda nejprve nerozhodnout, která data zpracovat a dále využít a která ne (Akhgar et al. 2015). Vhodným konkrétní ukázkou dat s nejistou věrohodností mohou být například údaje čerpané ze sociálních sítí.

Dalším rozšíření původního modelu charakterizujícího Big Data je vlastnost reprezentující vysokou hodnotu (value), kterou tato data pro firmy, které je vlastní, představují (Demchenko 2013). Shromažďováním a analýzou obrovských objemů dat z mnoha různých zdrojů a ve všech možných formátech s sebou totiž přináší možnost vydolovat z nich množství velmi hodnotných informací, které z klasických dat získat nelze.

Výše zmíněný výčet vlastností Big Data však stále není definitivní a postupem času přibývají další charakteristiky. Setkat se tak můžeme s modelem „6V", „7V", ale dokonce i „11V" (Buyya et al. 2016). Často zmiňovanou vlastností je například nestálost (variabilita) dat (z anglického variability). Ta poukazuje na neustálou změnu, vývoj a celkovou dynamičnost těchto dat. Na rozdíl od různorodosti, která je spojena s mnoha datovými formáty a jejich strukturou, souvisí variabilita s různým významem v sémantickém smyslu a jeho případnou změnou v čase (Akerkar 2013). Stejně tak jako může mít jedno obyčejné slovo více smyslů, mohou i stejná data v různém kontextu

představovat vždy něco jiného a navíc měnit svůj význam v závislosti na čase. V obecné rovině lze poté říci, že variabilita poukazuje na velké množství neznámých v dané datové sadě (Buyya et al. 2016).

Příkladem dalších vlastností spojovanými s Big Data jsou poté platnost a pomíjivost. Platnost (validity) poukazuje na skutečnost, že je důležité zabývat se nejen otázkou přesnosti a věrohodnosti dat, ale také ověřením jejich vhodnosti pro zamýšlené použití (Hurwitz et al. 2013). Příkladem pro lepší vysvětlení z reálného života je oblast zdravotní péče, kdy klinická studie může úzce souviset s onemocněním pacienta, nicméně lékař nemůže převzít její výsledky a postupy, aniž by byla ověřena právě její platnost. Pomíjivost (volatility) se poté vztahuje k otázce, jak dlouho jsou data využitelná a jak dlouho by měla být uložena (Hurwitz et al. 2013). Tento problém vzniká především v důsledku velkého objemu dat a jeho rychlého nárůstu. Pro mnohé společnosti není možné stále navyšovat kapacitu úložišť, a pokud si navíc uvědomíme, že mnoho informací slouží pouze pro rychlé analýzy, je více než důležité rozhodnout, která data jsou relevantní pro pozdější analýzy, a tudíž která shromažďovat, a která ne.




Obr. 1 Základní vlastnosti charakterizující Big Data.

I přesto, že se lze setkat ještě s dalšími vlastnostmi Big Data, například v (Borne 2014), lze výše zmíněné (shrnuté na obrázku 1) považovat za ty nejdůležitější. Přičemž platí, že základní „3V" model je většinou všeobecně přijímán, zatímco ostatní jsou navrhovány různými autory vždy v jiném počtu, případně i kontextu, a často se setkávají s kritikou (Li et al. 2016). Kromě toho charakteristiky začínající anglickými ekvivalenty

na písmeno „V“ nejsou jediné. Například (Suthaharan 2014) přichází s modelem „3C“ zahrnující kardinalitu, kontinuitu a složitost (cardinality, continuity, complexity). Nicméně důležitější než popis dalších vlastností je skutečnost, že se všechny zaměřují pouze na samotná data, zatímco pojem Big Data se váže i k architekturám a technologiím, které s nimi pracují (Carter 2011). Právě jejich vzestupem a možnostmi se stala Big Data v poslední době tak populární, neboť objemné datové sady s vlastnostmi popsanými výše tu byly i dříve, nicméně chyběly nástroje pro jejich efektivní zpracování a následné využití. Jako příklad lze jmenovat open-source framework Apache Hadoop určený pro zpracování velkého množství nestrukturovaných a distribuovaných dat, nebo NoSQL databázové systémy sloužící ke skladování a manipulaci s těmito daty.

Závěrem lze tedy říci, že pojem Big Data je stále dost nejednoznačný a existuje více teorií, jak ho definovat a popsat. Jeho nejdůležitějším aspektem jsou samotná data, strukturovaná i nestrukturovaná a s masivním objemem, pro která platí, že je nelze snadno zachycovat, skladovat, manipulovat s nimi, analyzovat, spravovat a prezentovat tradičním software, hardware a databázovými technologiemi v rozumném čase (Li et al. 2016). Kromě této obecné definice lze data charakterizovat podrobněji pomocí výše zmíněných vlastností, přičemž objem s jeho vysokým růstem, rychlost, s jakou je potřeba data zpracovat, a jejich různorodost z pohledu strukturovanosti (tedy „3V“ model) jsou považovány za ty nejdůležitější. Další vlastnosti poté spíše poukazují na problémy a skutečnosti, které s sebou Big Data mohou přinášet, a nad kterými je potřeba se před jejich zpracováním zamyslet. Nicméně není pevně stanoveno, zda musí určitá datová sada splňovat více nebo jen jednu zmíněnou vlastnost, a především v jaké míře, aby se dala označit jako Big Data. Definice a popis tohoto pojmu se poté liší i v závislosti na technologickém, průmyslovém, výzkumném, akademickém nebo jiném pohledu (Chen et al. 2014) a kromě nejčastějšího datově orientovaného přístupu lze na něj nahlížet i jako na technologie, architektury, aplikace, ale i jako na výzvu nebo pouhý marketingový pojem poukazující na jeho častou kritiku.

## 2.2 Prostorová Big Data

Pro mnoho datových sad, které se dají zařadit do oblasti Big Data, je základním prvkem poloha. Tedy určení vztahu objektu k určitému místu v prostoru, nejčastěji pomocí souřadnic v předem definovaném prostorovém referenčním systému (Li et al. 2016). Bez ní

by byly soubory dat méně hodnotné, v extrémních případech i bezcenné, jelikož je zásadním prvkem pro mnohé analýzy, hledání nových vzorů či trendů nebo pro celkové porozumění daného jevu. Pokud navíc připustíme známou frázi, že přibližně 80 % všech dat obsahuje prostorovou složku, viz diskuse (Dempsey 2012), je poté více než důležité umět s těmito daty pracovat a to i v případě, kdy mají vlastnosti Big Data. Jejich zdrojem může být například pozemní mapování, fotogrammetrie, dálkový průzkum Země nebo v poslední době stále více se rozvíjející laserové skenování, mobilní mapování, geolokační senzory, globální navigační družicové systémy (GNSS), crowdsourcing a práce dobrovolníků v oblasti prostorových dat (VGI), ale i příspěvky na sociálních sítích, ke kterým uživatelé připojili svoji polohu. Je nutné si také uvědomit, že ještě několik let zpět byl proces sběru prostorových dat založen pouze na technicky složitých a drahých přístrojích vyžadujících často speciální postupy měření a odbornou obsluhu. Dnes je však naopak možnost jejich sběru, například pomocí GPS, dobře dostupná i v běžných zařízeních, jako jsou chytré telefony. Ty dokáží získat prostorové informace s dostačující přesností, navíc jsou malé, snadno ovladatelné a často jsou schopny poskytovat údaje o poloze i bez interakce uživatele.

Prostorová složka však také vyžaduje specializované funkce, techniky a algoritmy sloužící pro jejich ukládání, analyzování a zpracování (Eldawy a Mokbel 2015). Konkrétně se jedná především o podporu prostorových datových typů, indexů a operací. A právě ty ve většině nástrojů a infrastruktur pro Big Data chybí. V praxi je tak často potřeba zacházet s těmito daty jako s neprostorovými nebo napsat nad samotným systémem sadu vlastních funkcí pro jejich podporu (Eldawy a Mokbel 2015). Taková řešení však mají ve většině případů za následek ztrátu výhody a potenciálu prostorové složky.

Ukládání, dotazování a analýza prostorových, případně časoprostorových, Big Data se obecně od většiny jiných dat liší především kvůli specifickým datovým formátům a informacím, které v sobě obsahují. Typy informací lze rozdělit na prostorové (poloha, tvar, vztah k jiným objektům), popisné (další vlastnosti daného objektu, tj. atributová data jako například délka linie nebo rok pořízení) a případně i časové (dynamické vlastnosti objektu). Na formáty lze poté tradičně nahlížet jako na rastrové a vektorové. Vektorové se typicky skládají z bodů, linií a polygonů, rastrové poté z jednotlivých bodů (pixelů) a jejich klasickým příkladem jsou satelitní snímky. Nicméně toto základní rozdělení vychází především z potřeb geografických informačních systémů (GIS) a plně neodráží

realitu prostorových Big Data. Ta jsou typická svoji různorodostí jak z pohledu strukturovanosti, tak právě z pohledu formátů. Lze se tak setkat i s grafovými daty či sítěmi (Lee a Kang 2015), mračnem bodů jako výsledkem měření LIDAR, ale i textově založenými soubory například ze sociálních sítí a aplikací či logů z různých zařízení obsahující kromě jiných informací i polohu (Olasz a Nguyen Thai 2016).

Prostorová Big Data tedy představují důležité téma k diskusi a díky svému potenciálu a možnostem využití je velice výhodné s nimi umět pracovat a získávat z nich informace. Čtvrtá kapitola této práce je proto zaměřena na možnosti a přístupy řešící jejich vizualizaci. Ta totiž představuje člověku přirozený způsob, jak z takových dat získávat nové poznatky a dosud neznámé skutečnosti. Ty lze poté uplatnit v široké škále oblastí, jako je například environmentálního plánování, krizový management, zdravotnictví nebo doprava. Dalšími a konkrétnějšími způsoby aplikací se poté zabývá například (Lee a Kang 2015) nebo (Li et al. 2016).

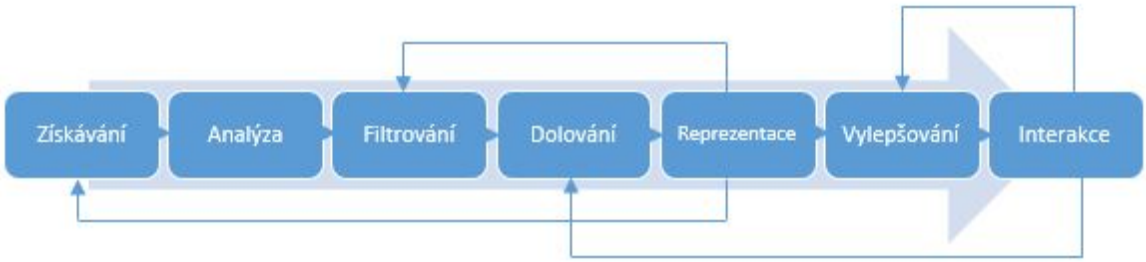
# 3. Proces vizualizace

V souvislosti s dalšími vlastnostmi Big Data je také často zmiňován termín vizualizace, popř. vizuální analytika jako další aspekt, který je potřeba brát v úvahu, pokud chceme s Big Data pracovat a získávat z nich informace. V některých zdrojích je dokonce vizualizace součástí již zmíněných modelů začínajících na písmeno „V“, např. v (DeVan 2016) nebo (Rijmenam 2016). Důvodem je především fakt, že i menší množství dat v nestrukturované formě je pro člověka nečitelné, a vizualizace tak představuje efektivní způsob jak datům porozumět, popř. zjistit, co vlastně obsahují. Zatímco tabulky a jiné textově, popř. číselně, založené soubory jsou vhodné pro hledání konkrétních hodnot, zobrazuje vizualizace spíše celkový pohled na data se vztahy a souvislostmi pro jednu nebo více proměnných. Jinými slovy je jejím cílem pochopení zkoumaných jevů a vniknutí do problému. K tomu využívá různé druhy a kombinace grafů, diagramů, schémat, obrázků, informační grafiky a v případě existence prostorové složky i map, digitálních modelů terénu či povrchu nebo jiných 3D modelů. Data jsou poté pro uživatele srozumitelnější a je možné je dále využít pro konkrétní analýzy. Kromě toho slouží vizualizace i k získávání nových informací a dosud neznámých skutečností (Li et al. 2016). Lze ji však také aplikovat i v pozdějších stádiích procesu zpracování a využití dat, například při prezentování výsledků analýz nebo pro další výzkum.

Ačkoliv vizualizace nemusí být z technologického hlediska nejnáročnější, představuje v procesu využití dat zcela zásadní krok, neboť je schopna převést jejich obrovské množství do člověku srozumitelné podoby. Mnoho společností a projektů v ní proto vidí budoucnost analýz Big Data (Olshannikova et al. 2015), což vede k vývoji nových vizualizačních technik, metod a nástrojů. Podle zprávy (SAS Institute 2013) dokonce 98 % společností, které jsou úspěšné v oblasti analýz Big Data, používá ve svých řešeních právě vizualizaci.

Samotný proces zpracování a vizualizace dat lze poté formalizovat podle (Fry 2007) do 7 základních kroků, viz obrázek 2. Prvním je získávání dat. Tedy nutnost disponovat podklady v dostatečném množství a kvalitě. Následuje analýza, jejímž cílem je pochopení předkládaných dat a jejich následná transformace do strukturované a nejlépe i strojem snadno zpracovatelné podoby. Na ni navazuje filtrování, které odstraní všechna data kromě těch, kterým se chceme věnovat. Často je

využito tzv. vícestupňové filtrování omezující data podle zadaných kritérií postupně. Čtvrtou fází je poté dolování. Zde se aplikují metody ze statistiky, rozeznávají se základní vzory nebo se data zasazují do matematického kontextu. Tento krok tedy umožnuje získat základní znalosti a informace z dat ještě před jejich reprezentací. Konkrétním jednoduchým příkladem může být procházení souboru a hledání minimální a maximální hodnoty zeměpisné šířky a délky pro následné zobrazení datové sady ve správné oblasti a v optimálním měřítku. Další krok, reprezentace, poté slouží k převedení získaných dat a informací do formy, která bude vizuálně snadno pochopitelná. Patří sem tedy rozhodnutí, zda se bude jednat například o graf nebo mapu a zda tato mapa bude mít podobu kartogramu nebo heatmapy (teplotní mapy). Následuje fáze vylepšování zabývající se vizuální kvalitou celého výstupu, tj. jeho čitelností, atraktivitou a srozumitelností. Konkrétně se může jednat o aplikování teorie barev, tedy například volbu, jaká barva na mapě bude reprezentovat jakou hodnotu. Posledním krokem je poté interakce. Ta přináší uživateli možnost manipulovat s daty, vybírat jejich podmnožiny, popř. řídit, které vrstvy budou viditelné a které ne.




Obr. 2 Proces vizualizace dat podle (Fry 2007).

Výše zmíněné kroky v procesu vizualizace dat však představují pouze obecný přístup a nelze je v každém případě otrocky následovat. Zda je potřeba je projít všechny nebo jen nějaké z nich, záleží vždy především na konkrétních datech, jejich podobě a cíli vizualizace (Fry 2007). Kromě toho mohou jednotlivé fáze zpětně ovlivňovat ty předchozí (viz obrázek 2), což vede k propojení celého procesu. Výsledek vizualizace je tedy často produktem iterací jednotlivých kroků a jejich neustálého vylepšování.

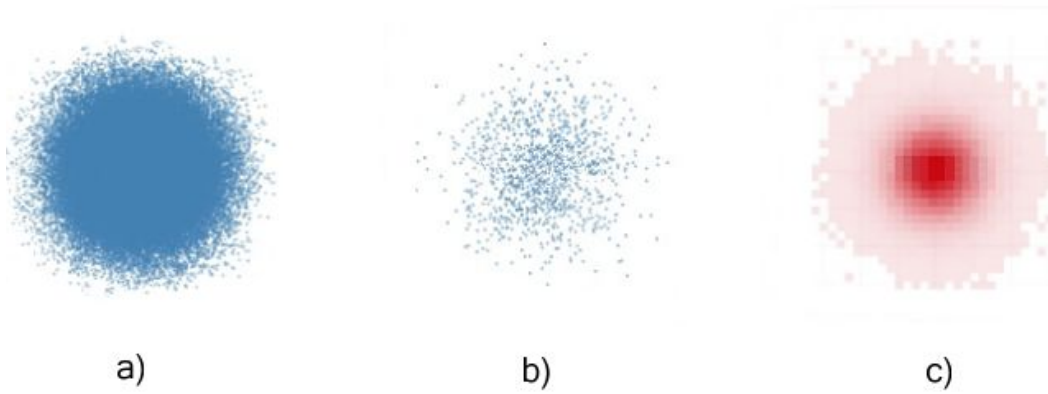
Je nutné si však také uvědomit, že vizualizace dat je jak věda, tak i umění (Aparicio a Costa 2014). Nejenže lze jednu datovou sadu vizualizovat různými způsoby, což vede k získání zcela rozdílných výsledků, ale i již vytvořená jedna konkrétní vizualizace může

být interpretována zcela odlišně jen proto, že se na daný problém díváme z různých pohledů. Tato skutečnost s sebou přináší poměrně vysoké nároky na znalosti a zkušenosti autora vizualizace, kterému by měli ideálně pomáhat odborníci i z jiných z oblastí. Příkladem mohou být počítačoví experti, kteří si poradí s filtrováním a zpracováním dat, sociologové formulující základní hypotézy a nápady, nebo designéři, popř. grafici, pomáhající s převedením výsledků do pochopitelné a aktraktivní formy (Černý 2013). Vizualizace tak nepochybně představuje riziko ve špatném a především nepřesném zpracování, což může vést k zavádějícím nebo až zcela nesmyslným výsledkům.

Dalším aspektem, který je potřeba brát v úvahu během tvorby vizualizace, jsou lidské percepční a kognitivní možnosti. Kromě aplikování obecných teorií, jako je například schopnost rozeznávat lépe rozdíly v délce čar než rozdíly ve velikosti ploch (Few 2004), se musí brát ohled na uživatelovo vnímání hlavně při rozhodování o počtu vykreslovaných prvků. Především v souvislosti s prostorovými Big Data totiž není efektivní zobrazit jeden prvek jako jeden samostatný bod na mapě. Ačkoliv je tento přístup velice často používaný, má za následek jejich i mnohonásobné překrývání. To sice nemusí být s ohledem na cíl vizualizace a informaci, kterou má podávat, vždy považováno za nevýhodu, nicméně ve většině případů vede k naprosté nepřehlednosti a nečitelnosti výsledné práce. Kromě toho není často možné tento postup aplikovat i kvůli technické stránce s ohledem na omezenou velikost a rozlišení displejů. Alternativou je poté redukce vykreslovaných dat. Do této oblasti spadá seskupování prvků na základě jejich prostorové nebo sémantické blízkosti do přiměřeného počtu shluků (též nazýváno shluková nebo clusterová analýza) viz obrázek 3b), nebo rozdělení prostoru na disjunktní oblasti, ve kterých jsou data následně agregována, obrázek 3c). Kromě těchto obecných postupů reprezentujících data jako diskrétní hodnoty, lze využít i techniky, které zobrazují distribuci dat spojitě, například heatmapy (Ježek et al. 2017). Ačkoliv však všechny tyto postupy vedou k čitelnější a pro lidské vnímání a poznávání vstřícnější vizualizaci, mohou na druhou stranu vést i ke ztrátě určitých informací. Vlivem shlukování či agregace totiž dochází k eliminaci zajímavých struktur a odlehlých hodnot (Liu et al. 2013).

V procesu vizualizace je tedy nutné rozlišovat dva rozdílné přístupy redukce dat. První je spojený s analýzou a filtrací, která snižuje objem datové sady ještě před jejím samotným vykreslením, případně výpočtem. Druhý poté souvisí s omezením počtu zobrazovaných prvků. To znamená, že jsou do vykreslení i výpočtu zahrnuty, jen jsou

následně kvůli jejich velkému počtu agregovány nebo shlukovány. Typicky je například zobrazen znak, který reprezentuje určité množství prostorově či sémanticky si blízkých bodů. Obecně by měl proces vizualizace zahrnovat oba dva přístupy. Tedy eliminaci jak dat, tak i počtu zobrazovaných prvků. Splnění tohoto přístupu lze považovat za základní předpoklad vedoucí k vytvoření přehledného a uživatelsky přívětivého výsledku.




Obr. 3 Ukázka redukce zobrazovaných dat: a) původní data - každý prvek zobrazen jako samostatný bod, b) redukce dat do shluků na základě jejich prostorové blízkosti, c) rozdělení prostoru na disjunktní oblasti, ve kterých jsou data agregována. (Kandel a Heer 2013).

Vizualizace spojená s prostorovými Big Data tak musí řešit především dva hlavní problémy. První se týká lidského vnímání a souvisí s výše zmíněnými přístupy pro snížení počtu zobrazovaných prvků. Zde je stěžejním úkolem především vybrat vhodnou konkrétní strategii pro jejich redukci. Ta musí být volena nejen s ohledem na cíl vizualizace, ale i tak, aby grafické zatížení mapy nebylo příliš velké a uživatel byl tak schopný z ní získat potřebné informace. Zároveň však nesmí například v případě bodových prvků dojít k příliš velkému snížení jejich počtu, což by mělo za následek ztrátu nebo výrazné zkreslení informace ohledně prostorového rozložení zkoumaných dat. Druhý problém poté vychází ze samotných vlastností Big Data. Týká se především rychlosti vykreslení vizualizace a případného interaktivního průzkumu dat uživatelem. Při dotazování velkých datových úložišť totiž vzniká vysoká latence, která má za následek dlouhou dobu načítání vizualizace a v případě interakce narušení její plynulosti. Tyto vlastnosti jsou pro uživatele samozřejmě nežádoucí a je tedy snahou je eliminovat. Nicméně zpracování, popř. výpočet, vizualizace Big Data je natolik složitý, že je nemožné ho dosáhnout v potřebném čase běžnými technologiemi.

Cíl této diplomové práce je tak zaměřen na druhý zmíněný problém spojený s vizualizací prostorových Big Data. Snahou je představit a zhodnotit jednotlivé možnosti jeho řešení a především i konkrétní nástroje a pokročilé techniky, popř. technologie, které lze pro různě velké datové sady použít.

První výše představený problém, tj. redukce zobrazovaných dat, nevychází přímo z problematiky Big Data, ale je spojen obecně s procesem vizualizace. Ve spojitosti s prostorovými daty se objevuje už i při zobrazení menšího množství prvků, pokud jsou například předkládány na mapě ve velkém měřítku. Této problematice je už věnována v literatuře dostatečná pozornost, a proto nebude v této práci řešena. S ohledem na prostorová data je rozebírána především ve spojitosti s metodami tematické kartografie, generalizací nebo s GISy. Jako příklad publikací zabývající se daným tématem nebo vytvářením konkrétních algoritmů pro redukci dat lze jmenovat (Elmqvist a Fekete 2009), (Pravda 2006) nebo (Anderson 2009). Obecné techniky pro zpracování vizualizace větších datových sad z pohledu shlukování, agregování apod. poté představuje například (Kocherlakota a Healey 2005)

Na tomto místě je také ještě dobré zmínit dělení vizualizačních technik založené na struktuře generovaného obrazu do dvou kategorií (Eldawy a Mokbel 2015). První představuje jednoúrovňový obraz, který je reprezentován jako jeden soubor s určitým rozlišením. Tento výstup je typický pro naprostou většinu nástrojů, je jednodušší z hlediska implementace, ale často nemůže zachytit všechny detaily. Při větším objemu dat je zde tedy naprosto zásadní jejich redukce. Druhou kategorií je poté víceúrovňový obraz skládající se ze sady jednotlivých obrazových dlaždic rozdělených do určitého počtu měřítkových úrovní. Tento přístup umožňuje uživateli zoomovat mezi jednotlivými vrstvami, a tím pádem vidět i více detailů v zájmové oblasti. Používá se právě především pro prostorová data jako mapové dlaždice, nad kterými jsou budovány moderní aplikace typu Google Maps[9]. Speciálně v souvislosti s prostorovými Big Data má tento způsob vizualizace značný potenciál, jelikož umožňuje uživateli přehledně procházet i obrovské množství dat. Nicméně jeho implementace je ve vizualizačních nástrojích pro Big Data víceméně ojedinělá, jedním z mála takových zástupců je například framework SpatialHadoop (viz kapitola 4.3).

[9] Google Inc. *Google Maps* [online]. Dostupné z: https://maps.google.com/

# 4. Možnosti vizualizace prostorových Big Data

Nejednoznačnost pojmu Big Data popsaná v druhé kapitole se odráží i do vizualizačních nástrojů a technik, které s nimi pracují. Jelikož není jasná shoda o tom, jak velká data musí být, aby se dala označit jako Big Data (respektive neexistuje ani žádné jiné přesné vymezení dalších vlastností), může s trochou nadsázky téměř každý přiřadit tento pojem k jakékoliv datové sadě, kterou považuje za objemnější, než pro něj bývá běžné. Často je však toto označení velmi sporné. I proto je pro účely této práce rozdělena vizualizace prostorových Big Data do dvou následujících kategorií.

První se týká "pravých" Big Data, které představují datové sady o velikosti až v řádu stovek gigabajtů, terabajtů a teoreticky i petabajtů[10] v mnoha různých formátech a s neustálým nárůstem těchto dat. Paměťová a výpočetní kapacita jednoho počítače nemůže samozřejmě ani za pomoci specializovaného software s těmito daty pracovat, a tudíž je třeba využít distribuované systémy a paralelní výpočty. Nutno podotknout, že s "pravými" Big Data se běžný člověk nemá většinou možnost setkat a jde spíše o záležitost velkých společností a výzkumných pracovišť jako je třeba NASA nebo Google. Ve spojitosti s prostorovými daty lze jako konkrétní příklad jmenovat ukládání a zpracování satelitních snímků. Tyto systémy se však samozřejmě používají i pro podstatně menší datové sady a vzhledem k jejich časté nekomerční licenci je tak teoreticky může použít každý.

Druhou kategorii poté reprezentují datové sady, jejichž objemy dosahují velikosti jednotek, maximálně desítek gigabajtů. Jelikož s nimi rovněž nejde pracovat běžnými nástroji a především v rozumném čase, lze je podle jedné z definic (Snijders et al. 2012) označit také jako Big Data. Nicméně jsou většinou reprezentovány jedním formátem a pevnou strukturou. Z toho důvodu je tedy označení Big Data už trochu diskutabilní. Typickým příkladem z pohledu prostorových dat jsou miliony bodově lokalizovaných prvků dané souřadnicemi a popisnými atributy ve formátech typu CSV, GeoJSON, KML nebo GML. Často tedy datové sady pokrývající celý svět. Tyto formáty jsou již běžné a nástrojů na jejich vizualizaci existuje celkem mnoho (například Leaflet). Nicméně si neporadí se zmíněným větším objemem dat. Proto je druhá kategorie zaměřena na pokročilé vizualizační nástroje, popř. jejich kombinace a speciální postupy,

[10] Americká společnost Yahoo! Inc. disponuje systémem Hadoop s více než 600 petabajtů dat (Bortnikov 2016). V tomto případě se však nejedná o data prostorová.

které zvládnou vizualizaci větších souborů, aniž by k tomu potřebovaly výpočetní kapacity více počítačů. Nutno podotknout, že je tento problém velice aktuální, neboť se s ním může setkat každý, kdo se obecně zabývá vizualizací prostorových dat.

Zmíněné rozdělení do dvou kategorií je příhodné i z pohledu složitosti na použití. Distribuované systémy představují poměrně komplexní záležitost a jejich nastavení vyžaduje mnoho znalostí, zkušeností a času. To samé platí i pro práci s nimi. Pro běžné uživatele je tedy toto řešení ve většině případů nedostupné a to i kvůli nemožnosti disponovat počítačovým clusterem, tj. seskupením více počítačů. Druhá kategorie je naopak řešitelná na jednom počítači, nástroje jsou jednoduché na instalaci a v prostředí internetu existuje poměrně mnoho návodů i rad, jak s nimi pracovat. Toto řešení je tedy výrazně jednodušší. Jeho zásadní nevýhodou je však existence horní hranice objemu dat, která lze takto vizualizovat.

Složitost obou kategorií se poté odráží i do možností, jak bude výsledná vizualizace vypadat a jaké bude mít případně vlastnosti. Distribuované systémy jsou vytvářeny s cílem zvýšit výpočetní rychlost a spoluprací všech jeho komponentů dosáhnout nějakého společného cíle. Tím mohou být různé analýzy nebo obecně manipulace s daty a snaha získat z nich potřebné informace. S ohledem na prostorová data je poté budována do systémů podpora především prostorových datových typů (např. bod, polygon), prostorových indexů (dlaždicový index, čtyřstrom) a prostorových operací (průnik, sjednocení). Nicméně podpora samotné vizualizace již v mnoha systémech chybí nebo je značně omezená. Často je tak jediným možným výstupem statický jednoúrovňový obraz, výjimečně poté i víceúrovňový ve formě mapových dlaždic (viz kapitola 3).

Druhou kategorii naopak reprezentují nástroje specializované přímo na vizualizaci, případně nástroje na její předzpracování a rychlejší práci se soubory. S ohledem na následnou prezentaci výsledků, většinou v prostředí internetu, je možné vytvářet interaktivní vizualizační aplikace. Ty navíc mohou být i dynamické, tj. měnit své parametry například v závislosti na volbě uživatele. Dále je možné vytvářet vizualizace v rastrové i vektorové podobě, nebo do výsledku zakomponovat i další informace, například zobrazit popisné atributy po kliknutí na daný bod. Možností a nástrojů je tedy v této kategorii podstatně více. Navržené dvě kategorie však nerozdělují vizualizaci prostorových Big Data na dvě disjunktní oblasti. Naopak je často výhodné je kombinovat. Typický příklad představuje využití distribuovaných systémů pouze na předzpracování dat.

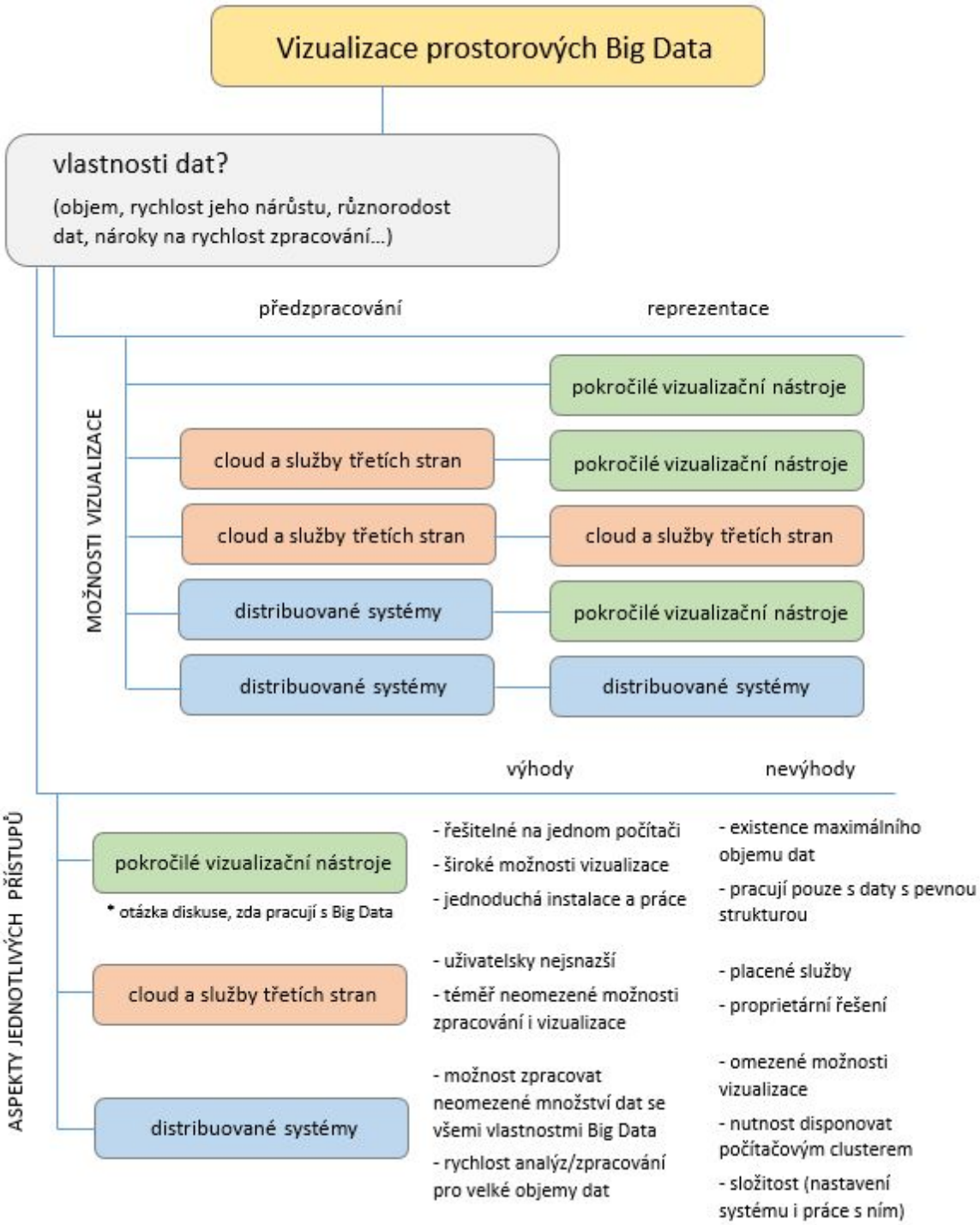
S ohledem na zmíněný proces vizualizace (viz kapitola 3) se tedy jedná jen o analýzu, filtraci a dolování. Následnou reprezentaci, vylepšování vizualizace, popř. možnosti její interakce, je poté výhodné zpracovat již v nástrojích z druhé kategorie, a tím tedy využít všechny jejich možnosti a výhody. Toto řešení však bohužel není vždy možné. Jedná se především o případy, kdy i po předzpracování musíme manipulovat s velkým objemem dat nebo kdy není možné potřebná data ze systému extrahovat do nějakého jednoduchého textově založeného formátu (například CSV), se kterým již umí nástroje z druhé kategorie pracovat. Často je tak kvůli stále velkému objemu možné vizualizovat pouze dílčí analýzy nebo výrazně menší podmnožiny dat.

Kromě dvou výše popsaných kategorií je nutné zmínit ještě jednu obecnou možnost jak řešit vizualizaci prostorových Big Data. Jde o využití cloudu (cloud computingu), který přestavuje poskytování služeb přes internet. S ohledem na téma této práce se jedná především o využití výpočetního výkonu poskytovatele cloudu. To znamená, že uživatel, který nedisponuje počítačovým clusterem, ale potřebuje zpracovat Big Data, může tímto způsobem využít distribuovaný systém třetích stran. V naprosté většině případů se však jedná o placenou službu, kdy uživatel neplatí za jednotlivé počítače v systému, ale za jejich využití.

S ohledem na vizualizaci prostorových Big Data lze cloud použít především dvojím způsobem. První zahrnuje propůjčení úložiště a výpočetního výkonu na skladování dat a na jejich dotazování. Uživatel tak pouze posílá dotazy (například ve formě SQL) a o jejich zpracování se starají systémy na straně poskytovatele cloudu. Ten většinou garantuje vysokou rychlost jejich vyřešení, která v naprosté většině případů spočívá právě ve využití distribuovaných systémů a paralelních výpočtů. Získaná data už poté uživatel může vizualizovat sám. Druhou možností je poté využití služeb třetích stran pro celý proces vizualizace prostorových Big Data. Poskytovatelé tedy zajistí nejen ukládání dat, manipulaci s nimi, potřebné výpočty, ale i jejich samotné vykreslení a to někdy i s možností této vizualizace ve formě interaktivní aplikace (viz podkapitola 4.2). Nicméně čím více služeb uživatel využije, tím samozřejmě více zaplatí.

Vizualizace prostorových Big Data lze tedy obecně řešit za pomoci distribuovaných systémů a paralelních výpočtů, cloudu a využití služeb třetích stran nebo pokročilými vizualizačními nástroji a technikami (viz obr. 4). Přitom platí, že lze jednotlivé přístupy kombinovat a to s ohledem na jejich využití pro předzpracování dat nebo jejich samotnou

reprezentaci. Dále platí, že cloud obecně využívá oba dva zbývající přístupy a jedná se tedy o jiné řešení pouze z uživatelova pohledu. Ten tímto využívá služby třetích stran a vyhýbá se tedy samotnému problému vizualizace. Toto řešení je pro něj samozřejmě nejsnazší, avšak zpoplatněné.




Obr. 4 Základní obecné přístupy vizualizace prostorových Big data s jejich aspekty.

Dále je důležité, že volba přístupu není závislá ani tak na uživateli jako spíše na povaze dat, s kterými hodlá pracovat. Zásadní jsou především jejich vlastnosti, jako je objem a rychlost jeho nárůstu, nároky na rychlost zpracování, jejich různorodost a případně i další vlastnosti popsané v druhé kapitole této práce. Pro pokročilé vizualizační nástroje navíc platí, že je otázkou sporu, zda lze datové sady, se kterými mohou pracovat, považovat za Big Data. Obecně se dá spíše říci, že pracují pouze s většími datovými sadami než je běžné (viz začátek této kapitoly a dále v kapitole 4.3). Obrázek číslo 4 poté výše zmíněné možnosti vizualizace přehledně shrnuje a poukazuje na jejich hlavní výhody a nevýhody. V následujících podkapitolách jsou dále všechny tři základní možnosti řešení blíže popsány a především představeny konkrétní nástroje, které lze využít.

## 4.1 Distribuované systémy a paralelní výpočty

Distribuovaný systém je obecně takový systém, ve kterém jeho jednotlivé softwarové a hardwarové komponenty umístěné na počítačích v síti spolu komunikují a koordinují své kroky prostřednictvím předávání zpráv (Coulouris 2011). Typicky se tedy jedná o skupiny počítačů spojené sítí, jejichž snahou je za pomoci jeden druhého dosáhnout společného cíle, který je jen obtížně nebo zcela neřešitelný samostatně. Celý systém se přitom navenek uživateli jeví jako jednotný celek. Dále je pro něj charakteristická nepřítomnost globálních hodin, nezávislost jednotlivých výpočetních entit (počítačů nebo uzlů) a absence sdílené paměti (Andrews 2000). Právě posledním bodem je ve většině případů odlišován od systémů paralelních. V distribuovaném systému má tedy typicky každý procesor svoji vlastní paměť, zatímco v paralelním využívá více procesorů jednu sdílenou. Slovo distribuovaný historicky odkazovalo na počítačové sítě, kde byly jednotlivé počítače fyzicky distribuovány v rámci určité geografické oblasti (Lynch 1996). Nicméně dnes se již tento pojem používá v širším významu, tedy i na autonomní procesy, které běží na stejném fyzickém počítači a komunikují mezi sebou prostřednictvím zpráv (Andrews 2000). I díky tomu se distribuované a paralelní systémy hodně překrývají, neexistuje mezi nimi přesně stanovená hranice a jeden konkrétní systém může být označen jako distribuovaný i jako paralelní. Paralelismus se přitom spíše týká dosažení konkrétního výpočtu tak rychle, jak je to jen možné, zatímco distribuovaný výpočet je obecnější a kromě rozdělení úlohy na více počítačů se zabývá i dalšími aspekty jako je konzistence

nebo dostupnost. Typickým příkladem je distribuovaný systém, ve kterém na jednotlivých počítačích nebo skupinách počítačů probíhají díky většímu množství procesorů, případně i jader, paralelní výpočty (Lynch 1996).

V souvislosti se zpracováním Big Data je nutné zmínit především softwarové frameworky Apache Hadoop a Apache Spark. Ačkoliv slouží každý k trochu jinému účelu, jsou v mnoha případech nástroje na zpracování a případnou vizualizaci prostorových Big Data postaveny právě nad nimi. V následujících několika odstavcích je proto zjednodušeně popsán princip jejich fungování, jelikož je víceméně platný i pro dále zkoumané vizualizační nástroje.

Apache Hadoop slouží obecně k ukládání a zpracování dat v distribuovaném prostředí. Je navržen tak, že může škálovat data na stovky či tisíce počítačů a tím využít všechny jejich výpočetní a diskové kapacity (Jain 2017). Jestliže tedy potřebujeme zpracovat Big data o určité velikosti, musíme disponovat odpovídajícím úložištěm a výpočetním výkonem. Toho docílíme právě spojováním více počítačů do volně vázaného seskupení, tzv. počítačového clusteru. Přičemž následným přidáváním uzlů můžeme objem zpracovávaných dat libovolně zvětšovat. Základní myšlenkou je tedy uložení dat na velké množství běžných počítačů, jejichž pořízení je mnohonásobně levnější než cena jediného superpočítače, který by díky obrovskému množství procesorů dosahoval stejných parametrů jako zmíněný celek spojených počítačů.

Framework Hadoop je sám o sobě napsaný v jazyce Java, dostupný pod open-source licencí (Apache License 2.0) a jeho první verze byla vydána v roce 2006[11] . Pro svoji správnou funkci potřebuje Java Runtime Environment (JRE) a standardně i nastavený zabezpečený komunikační protokol mezi jednotlivými uzly v clusteru, tj. Secure Shell (SSH). Jeho důležitou vlastností je předpoklad, že selhání hardwaru je běžný jev, který by měl být vyřešen automaticky samotným frameworkem. Funkčnost je tedy zachována i při výpadku jednoho nebo několika uzlů. Základem frameworku jsou 4 moduly - Hadoop Distributed File System (HDFS), Hadoop MapReduce, Hadoop Common a Hadoop YARN. První zmíněný představuje vlastní distribuovaný souborový systém, který rozděluje objemná vstupní data na části (bloky) a ty poté odesílá do uzlů v rámci clusteru. Zároveň je každý blok replikován na více uzlů, takže při selhání jednoho z nich nedojde ke ztrátě přístupu k datům. Kromě toho HDFS i data indexuje a uchovává si

[11] Viz archiv vydání Hadoop: http://archive.apache.org/dist/hadoop/core/

informaci, kde se nacházejí. Díky tomu je možné jejich následné zpracování. Tím se zabývá již další modul, programový model MapReduce, díky kterému je úloha rozdělena a zpracovávána paralelně na více uzlech. Typicky tak jeden z nich (nejčastěji master) přijme požadavek MapReduce od klienta a rozešle funkci Map všem ostatním uzlům clusteru, které provedou kód této funkce a vrátí výsledek zpět masteru. Jedná se tedy jak o distribuovaný, tak i paralelní algoritmus (Dean a Ghemawat 2004). Konečný výsledek je poté zjištěn výpočtem z dílčích výsledků. Funkce Map se tak stará o rozdělení úlohy a funkce Reduce o spojení výsledků. Výpočet je tedy z velké části přesunut k datům, čímž se značně redukuje potřeba přenosu velkých objemů dat po síti. I zde je přitom brán ohled na možné výpadky, a proto může být stejná úloha vykonávána na několika uzlech.

Co se týká posledních dvou modulů, Hadoop Common obsahuje knihovny a utility, které jsou potřeba pro ostatní moduly, a zároveň skripty pro samotné spuštění celého frameworku. Hadoop YARN poté slouží jako řídící platforma pro plánování úloh a správu výpočetních zdrojů.

Nicméně pojem Hadoop se dnes již nevztahuje pouze na čtyři výše zmíněné moduly, ale je jím často označován celý ekosystém frameworků, nástrojů a jiných softwarových balíčků, které jsou budovány nad samotným Hadoopem nebo které s ním velice úzce spolupracují (Jain 2017). Jednotlivé nástroje si však nekonkurují, ale naopak se vzájemně doplňují. Uživatel tak typicky nevyužívá pro řešení svého konkrétního problému jen jeden nástroj, ale většinou jejich vhodnou kombinaci. Celý ekosystém je přitom zaměřen na velkoobjemná data a Big Data.

Jedním z frameworků patřících do zmíněného ekosystému je i Apache Spark. Jeho hlavním rozdílem oproti Hadoopu je, že se nezabývá distribuovaným uložením dat, ale pouze jejich zpracováním. Spark lze tedy použít i samostatně, nicméně v sobě nemá zabudovaný žádný souborový systém, takže je potřeba nějaký integrovat. Takovým systémem může být například MapR File System, OpenStack Swift nebo Amazon Simple Storage Service. Nejčastěji je však využit právě HDFS od Hadoopu. Spark tím tedy pouze nahrazuje MapReduce svým vlastním modelem pro zpracování a analýzy. Jeho hlavní výhodou je především rychlost. Zatímco MapReduce postupuje v krocích, které postupně zapisuje na disk, Spark za pomoci paměti operuje s celým souborem dat naráz (Noyes 2015).

Kromě Sparku jsou důležitou součástí Hadoop ekosystému i distribuované NoSQL databázové systémy jako je Apache HBase, Apache Cassandra nebo Apache Accumulo. Nicméně sem patří i mnoho dalších nástrojů[12] z jiných oblastí, než jsou jen databáze. Příkladem může být podpora paralelního programování (Apache Pig), otázka vyššího zabezpečení (Apache Sentry) nebo datový sklad nad HDFS umožňující dotazování založené na SQL (Apache Hive). Jako alternativu k systému Hadoop a nástrojům pod hlavičkou Apache lze poté jmenovat například open-source High Performance Computing Cluster (HPCC) využívající vlastní distribuovaný souborový systém Thor a zpracovávající engine Roxie.

Na tomto místě je také nutné odkázat na práci se zmíněnými systémy na Katedře geomatiky ZČU pod hlavičkou evropského projektu OpenTransportNet[13]. V rámci něho je vytvářena otevřená dopravní mapa, přičemž pro výpočet dopravní intenzity pro EU je jako hardwarová platforma využita právě kombinace HDFS a Apache Spark (Jedlička et al. 2017). Použity jsou přitom výpočetní zdroje Národní Gridové Infrastruktury MetaCentrum [14] od sdružení CESNET. Projektem OpenTransportNet je dále také podporována tvorba nástroje pro pokročilou vizualizaci geografických dat, knihovny WebGLayer, která je podrobněji představena v kapitole 4.3.

Pro dále zmíněné nástroje pro vizualizaci prostorových Big Data je zcela zásadní, že nemohou být používány samostatně. Pro svoji funkci potřebují obecně především distribuovaný souborový systém, engine na zpracování a analýzy a správce celého clusteru (jako je například Hadoop YARN). Zmíněné moduly tedy nabízí Apache Hadoop, který všechny tyto komponenty již obsahuje. Nicméně v mnoha případech je výhodnější nebo přímo nutné využít kombinaci více nástrojů. Typicky je tak ještě integrován Apache Spark nahrazující část Hadoopu, nebo NoSQL distribuované databáze, které naopak části Hadoopu využívají. Dále je nutné si uvědomit, že zkoumané vizualizační nástroje tyto již složité kombinace rozšiřují a jejich použití s sebou přináší poměrné vysoké nároky na sestavení celého systému a především na jeho správnou konfiguraci. Kromě toho je samozřejmě nutné fyzicky disponovat počítačovým clusterem. U každého dále představeného vizualizačního nástroje tedy není vzhledem k výše zmíněnému principu

[12] Přehled nástrojů, které lze zařadit do ekosystému Hadoop, je v přehledné tabulce dostupný například na: https://hadoopecosystemtable.github.io/
[13] http://project.opentransportnet.eu/
[14] https://www.metacentrum.cz/cs/

funkce uvedena celá hierarchie softwaru, který ke své práci potřebuje, ale vždy pouze ten, do kterého je daný nástroj přímo zabudován či který rozšiřuje.

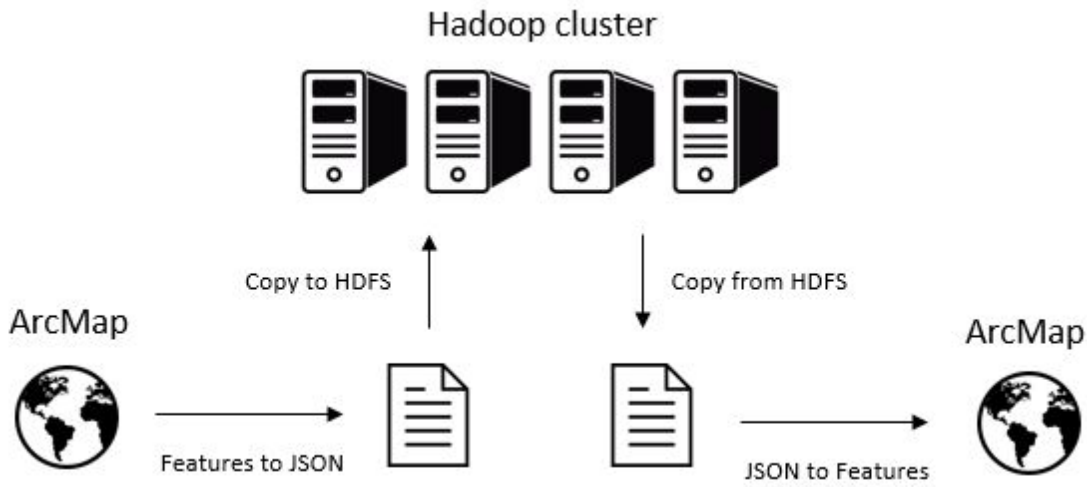
Prvním konkrétním nástrojem pro vizualizaci prostorových Big Data je open-source framework SpatialHadoop. Ten rozšiřuje samotný Hadoop, nicméně ne tak, že by byl instalován nad ním, ale je zabudován přímo do jeho jádra (Eldawy a Mokbel 2015). Tento přístup s sebou přináší mnoho možností, jak s prostorovými daty efektivně a rychle pracovat. SpatialHadoop tak podporuje prostorové datové typy, které může uživatel libovolně rozšiřovat, popř. definovat úplně nové, dále mnoho prostorových indexů i prostorových operací. Z pohledu uživatele je také důležité, že nabízí alespoň základní návody a konkrétní ukázky, jak s prostorovým rozšířením pracovat. Kromě toho jsou na oficiálních stránkách k dispozici i ukázkové datové sady z oblasti Big Data, které je možné pro naučení s nástrojem volně využít. Za nevýhodu lze naopak považovat stručnou dokumentaci pouze ve formě Java tříd. Co se týká samotné vizualizace, nabízí SpatialHadoop jak jednoúrovňový, tak i víceúrovňový výstup. Navíc však umožňuje uživateli definovat i nový typ vizualizace a tím ji přizpůsobit svým potřebám. Toho je možné docílit rozšířením abstraktní třídy "Rasterizer", konkrétně úpravou jejích pěti hlavních metod zajišťující vizualizační logiku. Následně je tato uživatelem upravená třída akceptována buď třídou "SingleLevelPlot" nebo "MultilevelPlot" (generování jednoúrovňového nebo víceúrovňového obrazu), která pomocí MapReduce vytvoří odpovídající výstup, standardně ve formátu PNG. Tímto způsobem je možné vytvořit například heatmapu, která již takto naprogramována byla a od té doby je její předdefinovaná třída součástí zdrojového kódu frameworku. Nicméně řešení s vlastní úpravou vyžaduje pokročilé znalosti nejen samotného nástroje, ale především programování v jazyce Java. I tak lze ale SpatialHadoop považovat za velice užitečný a to především s ohledem na možnou víceúrovňovou vizualizaci, která u podobných nástrojů většinou chybí.

Dalším nástrojem je poté GeoTrellis. Ten využívá Apache Spark, je psaný v jazyce Scala a dostupný opět pod open-source licencí. Umožňuje především práci s prostorovými rastrovými daty a to díky širokým možnostem mapové algebry. Obecně se jedná o matematické operace, které se vykonávající buď na jedné, nebo více vrstvách a jejichž výstupem je vždy vrstva nová. Tu je samozřejmě možné nejen použít v dalších analýzách, ale také vykreslit. GeoTrellis nabízí možnost jednoúrovňového i víceúrovňového výstupu,

standardně ve formátech PNG nebo JPEG. Dalšími užitečnými funkcemi v rámci rastru může být projekce z jednoho souřadnicového systému do jiného, ořezávání, seskupování nebo rozdělování, vytváření histogramů s četností jednotlivých hodnot, interpolace, tvorba stínovaného reliéfu nebo vzdáleností analýzy. Možností nabízí tento nástroj opravdu mnoho. Dále umožňuje pracovat i s vektorovou reprezentací dat, a to především díky podpoře formátu GeoJSON. Opět disponuje funkcemi pro manipulaci s takovými daty (geometrické operace, interpolace, reprojekce apod.), nicméně již nezajišťuje jejich vizualizaci. Jediným řešením je v této situaci převod vektoru na rastr, který již GeoTrellis vykreslit umí. Zajímavými zatím pouze experimentálními moduly jsou poté možnosti integrace dále zmíněných nástrojů GeoMesa a GeoWave jako datových zdrojů a podpora vektorových mapových dlaždic. Lze tedy předpokládat další vývoj tohoto nástroje, který opět přinese nové možnosti nejen pro práci s prostorovými daty, ale i pro vizualizaci.

Vzhledem k rozšířenosti produktů ArcGIS od společnosti Esri je také dobré zmínit sadu nástrojů GIS Tools for Hadoop. Tu tvoří čtyři projekty - Esri Geometry API for Java, Spatial Framework for Hadoop, Geoprocessing Tools for Hadoop a stejnojmenný GIS Tools for Hadoop. První zmíněný zavádí do Hadoopu podporu prostorových datových typů, indexů, operací, a umožňuje tak v jazyce Java vytvářet vlastní MapReduce aplikace pro analýzy nad prostorovými daty. Lze ho tedy použít i jako samostatnou knihovnu. Spatial Framework for Hadoop poté obsahuje uživatelsky definované funkce (UDF), kterými rozšiřuje Apache Hive a které jsou vytvořeny speciálně pro první zmíněný nástroj Esri Geometry API for Java. Umožňuje tak vytvářet dotazy za pomoci Hive Query Language (HQL), který je velice podobný SQL. Jinými slovy nabízí uživateli možnost vyhnout se psaní složitých MapReduce programů tím, že mu umožní pracovat s Esri Geometry API for Java pomocí jednoduššího HQL. Třetí projekt, Geoprocessing Tools for Hadoop, poté umožňuje jednak propojit Hadoop a ArcMap a dále i převádět prvky z a do formátu JSON. Jeho použití je z pohledu uživatele velice jednoduché, neboť jsou jeho funkce dostupné jako toolboxy v programu ArcMap. Typickým příkladem použití (viz obr. 5) může být postup, kdy jsou jednotlivé prvky v prostředí ArcMap převedeny do JSON (funkcí “Features to JSON”), dále je tento soubor vložen do distribuovaného souborového systému (funkcí “Copy to HDFS”), zde proběhne potřebná analýza za pomoci prvních dvou zmíněných nástrojů (tj. analýza která by v prostředí ArcMap trvala příliš dlouho nebo by kvůli nedostatečným výpočetním parametrům vyústila v pád programu)

a výsledek je opět vrácen jako soubor JSON ("Copy from HDFS") a konvertován do jednotlivých prvků ("JSON to Features"), s kterými již ArcMap běžně pracuje. Ve většině případů je však potřebná pouze druhá část, tj. máme data v HDFS, na která aplikujeme potřebnou analýzu, případně je jinak předzpracujeme, a výsledek chceme zobrazit v programu ArcMap. Ten již nabízí širokou nabídku jejich vizualizace od vektorové statické mapy až po mobilní aplikace nebo zveřejnění výsledků v rámci ArcGIS Online či Server. Poslední ze čtyř projektů, GIS Tools for Hadoop, poté integruje 3 výše zmíněné do jedné nástrojové sady a zároveň obsahuje ukázky a instrukce, jak je využít. Uživatel si tak díky němu může otestovat správné nastavení celého systému ještě před tím, než ho bude využívat pro vlastní řešení.




Obr. 5 Princip využití GIS Tools for Hadoop pro vizualizaci dat přes ArcMap.

Celá sada nástrojů GIS Tools for Hadoop tak představuje efektivní, s ohledem na ostatní nástroje poměrně jednoduchou a uživatelsky přívětivou cestu, jak data z HDFS vizualizovat. I přesto že samotná vizualizace probíhá přes program ArcMap, je nutné zde toto řešení zmínit. Za hlavní důvod lze považovat mnoho možností, které tento program v rámci vizualizace nabízí, ale i celkovou rozšířenost produktů ESRI. Důležitá je také existence velice podrobných návodů, jak celý systém nastavit a správně používat. Nutno však podotknout, že na rozdíl od samotné sady nástrojů GIS Tools for Hadoop, kterou lze zdarma využívat díky open-source licenci, jsou již produkty ArcGIS proprietární a placené.

Dalším nástrojem, který sám o sobě nepodporuje vizualizaci, ale nabízí možnosti, jak jí dosáhnout, je GeoMesa. Jedná se o open-source, distribuovanou,

prostorově-temporální databázi vytvořenou v jazyce Scala, která může standardně rozšiřovat NoSQL databáze Accumulo, HBase, Cassandra nebo systém na zpracování datových proudů Apache Kafka. Stejně jako předchozí nástroje podporuje GeoMesa prostorové datové typy a dále i prostorově-temporální index, který umožňuje vytvářet prostorové (a prostorově-temporální) dotazy. Výhodou je přehledná dokumentace s podrobným popisem včetně ukázek kódu a návody týkající se konkrétních problémů. Samotná vizualizace je poté řešena zásuvným modulem do GeoServeru, kterému je tak umožněno získávat data přímo z popisované databáze. Uživatel už poté jen v prostředí GeoServeru nastaví parametry vykreslení a výsledky může dále publikovat například pomocí WMS nebo jiných standardů OGC. Velkou výhodou je možnost vytvářet interaktivní aplikace, kdy je nejčastěji použita již existující základní mapová vrstva typu OpenStreetMap a nad ní je vykreslena vrstva vlastních dat například ve formě bodů. Na ty je navíc možno kliknout a na základě dotazu na databázi GeoMesa zobrazit přes GeoServer další atributové informace. Důležité přitom je, že proces vykreslení probíhá přímo na serveru a není tedy nutné posílat objemná data po síti přímo k uživateli. Nicméně i tak je nutné si uvědomit, že zatímco data (a dotazy na ně) se týkají celého počítačového clusteru, samotná vizualizace probíhá pouze na jediném počítači, na kterém běží GeoServer. Pokud je tedy potřeba vykreslit obrovské množství dat najednou, je v závislosti parametrech serveru nutné počítat s vyšší latencí nebo i s možným nepovolením tohoto procesu ze strany GeoServeru na základě jeho nastavených limitů.

GeoMesa dále nabízí ještě jednu možnost řešení vizualizace. Obsahuje totiž modul GeoMesa Spark na provádění prostorových analýz za pomoci enginu Apache Spark. Z něho lze následně integrovat výsledek do interaktivního webového prostředí Jupyter Notebook, který umožňuje jeho vizualizaci (více o tomto nástroji v podkapitole 4.3). Nicméně tato cesta představuje pro uživatele poměrně složité řešení především s ohledem na nutnost instalace a správného nastavení celé řady nástrojů a systémů. Samotný Jupyter Notebook poté navíc tvoří vizualizaci na základě uživatelovo programování, nejčastěji v Pythonu, což klade další nároky na jeho znalosti. Z pohledu nástroje GeoMesa tak opět nejde o přímou podporu vizualizace, ale pouze o možnost propojení s jinými systémy, které ji již umožňují.

Dalším nástrojem pro vizualizace prostorových Big Data je Babylon. Ten přímo rozšiřuje GeoSpark, který zavádí podporu prostorových datových typů, indexů a operací

do systémů založených na Apache Spark. Pro dosažení vizualizace je tedy potřeba do sebe zakomponovat více nástrojů, nicméně ty jsou pro tento účel vytvářeny, takže je jejich nastavení mnohem jednodušší než například zmíněný proces vizualizace dat z programu GeoMesa přes Jupyter Notebook. Nástroj GeoSpark je navíc od verze 0.5.0 poskytován již s přímo integrovaným rozšířením Babylon. Tento vizualizační nástroj je navržen pro paralelní vykreslování obrazů s vysokým rozlišením. Jinými slovy tedy pro svoji práci využívá výpočetní sílu celého clusteru. Jeho specializace přímo na vizualizaci navíc umožňuje měnit parametry výstupu podle potřeb uživatele. Je tak možné nejen vykreslit data v základní podobě (body, polygony), ale i vytvořit heatmapu nebo kartogram. Samozřejmostí je vlastní volba barev a barevných stupnic. Dále Babylon nabízí obrazové filtry (rozmazání, protlačení, detekci hran, doostření), překrývání více mapových vrstev přes sebe a výstupy ve formátu PNG, JPEG nebo GIF. Kromě toho umožňuje uživateli rozšířit, popř. definovat, vlastní obrazové filtry, typy map a teoreticky i všechny ostatní funkce celého nástroje. Ačkoliv jsou tedy jeho výstupem obrazy pouze jednoúrovňového charakteru, lze Babylon díky svým mnoha možnostem považovat v oblasti vizualizace prostorových Big Data za velice užitečný. Jako jeho nevýhoda se dá poté označit dokumentace opět pouze ve formě Java tříd.

Dalším zkoumaným nástrojem je GeoWave. Ten standardně rozšiřuje databáze Accumulo a HBase a opět do nich přináší podporu práce s prostorovými daty. Je vytvořen nad sadou nástrojů GeoTools, což umožňuje snadnou integraci a propojení s jinými projekty a datovými zdroji, které jsou s ní kompatibilní. Stejně jako GeoMesa, která v sobě rovněž implementuje GeoTools, tak nabízí možnost vizualizace přes GeoServer. Kromě toho však obsahuje zásuvný modul i pro Mapnik. Ten slouží k vytváření nejen jednoúrovňových obrazů, ale i víceúrovňových mapových dlaždic a je používán například projektem OpenStreetMap nebo MapBox. Díky zmíněnému pluginu je tak možné nastavit Accumulo jako zdroj dat a vlastní styl a podobu mapy již upravit standardním způsobem přes Mapnik. GeoWave tedy v podstatě slouží k propojení distribuovaných výpočetních systémů se softwarem pro práci s prostorovými daty.

Celkově lze tedy vizualizaci prostorových Big Data za pomoci distribuovaných systémů a paralelních výpočtů hodnotit jako značně omezenou. Nástrojů existuje velmi málo a některé z nich nabízejí vizualizaci pouze přes propojení na jiný software (viz tab. 1). Nicméně je nutné si uvědomit, že první plná verze Hadoopu (v. 1.0.0) vyšla

teprve v roce 2011[15] a Sparku dokonce až v roce 2014[16]. Popisované vizualizační nástroje tedy vznikaly až poté, a jsou tak poměrně novou záležitostí, která se i přes svůj rychlý rozvoj k běžným uživatelům zatím příliš nedostala. Druhým důvodem ne tak častého využití zmíněných vizualizačních nástrojů jsou poté jejich vysoké nároky na znalost výpočetní techniky, celková náročnost použití a potřeba programování.

Tab. 1 Přehled nástrojů pro vizualizaci prostorových Big Data se základním popisem.

| **Název** | **Základ** | **Licence** | **Vizualizace** |
|---|---|---|---|
| SpatilaHadoop | Apache Hadoop | Apache Licence 2.0 | jednoúrovňový, víceúrovňový obraz<br>základní vykreslení (body, polygony), heatmapa<br>+ možnost vlastního rozšíření |
| GeoTrellis | Apache Spark | Apache Licence 2.0 | jednoúrovňový, víceúrovňový obraz<br>mapová algebra a mnoho dalších funkcí pro práci s rastry |
| ESRI GIS Tools for Hadoop | Apache Hadoop | Apache Licence 2.0 | přes ArcMap |
| GeoMesa | Apache Accumulo<br>Apache HBase<br>Apache Cassandra<br>Apache Kafka | Apache Licence 2.0 | přes GeoServer<br>přes Jupyter Notebook |
| Babylon (GeoSpark) | Apache Spark | MIT License | jednoúrovňový obraz<br>základní vykreslení (body, polygony), heatmapa, kartogram<br>obrazové filtry<br>+ možnost vlastního rozšíření |
| GeoWave | Apache Accumulo<br>Apache HBase | Apache Licence 2.0 | přes GeoServer<br>přes Mapnik |

[15] Apache Hadoop - novinky oledně vydaných verzí. Dostupné z: http://hadoop.apache.org/
[16] Archiv vydání Apache Spark. Dotupné z: https://archive.apache.org/dist/spark/

S ohledem na provedenou rešerši lze však konstatovat, že podporu prostorových dat a operací nabízí v distribuovaných systémech stále více nástrojů a rozšíření. Kromě výše popisovaných se jedná například o SpatialSpark, STARK, Postgres-XL nebo LocationSpark. Nicméně ty se zabývají pouze prostorovými daty a nenabízí již možnost jejich vizualizace.

Konkrétní nástroje zkoumané v této podkapitole jsou shrnuty v tabulce 1. Cílem této práce ani záměrem autora však není vytvořit seznam vizualizačních nástrojů pro prostorová Big Data, ale představit pouze ty nejdůležitější a především pouze ty, které lze uživatelem reálně využít. Kromě nich totiž existují i jiná řešení, často v raném stádiu vývoje, která jsou ve většině případů výsledkem práce výzkumných pracovníků a institutů. Tyto nástroje však spíše představují zajímavé technologické postupy, ale lze je jen obtížně použít uživatelem, který se nepodílí na jejich vývoji. Příkladem může být (Bronson et al. 2011).

Oblast vizualizace prostorových Big Data pomocí distribuovaných systémů a paralelních výpočtů je tedy nutné považovat za teprve se rozvíjející. Lze však předpokládat její stále vzrůstající význam, a to především díky obecnému nárůstu dat, potřebě prezentovat prostorová Big Data člověku srozumitelným způsobem a rozšířenosti open-source licence v celé této oblasti.

## 4.2 Využití cloudu a služeb třetích stran

Druhou možností, jak řešit problém vizualizace prostorových Big Data, je využití cloudu a obecně služeb třetích stran. Ty nabízejí uživatelům možnost pronájmu svých počítačových clusterů a tedy vytváření distribuovaných systémů za pomoci nástrojů z Hadoop ekosystému i bez nutnosti nákupu nových strojů. Uživatel se tak sice většinou nemusí starat o instalaci a konfiguraci celého systému, nicméně s ním i přesto musí umět pracovat, vytvářet dotazy a především sám zajistit výslednou vizualizaci. Pokud se však do jejího procesu nechce zapojit vůbec, může si objednat celé její zpracování od specializovaných společností. V takových případech se již nestará téměř o nic, ale musí na druhou stranu počítat s vyšší částkou, kterou bude potřeba za tyto služby zaplatit. V následujícím textu je tedy představeno několik poskytovatelů nabízející své počítačové clustery a úložiště k pronajmutí a dále i několik konkrétních zástupců zaměřujících se na zpracování celého procesu vizualizace prostorových Big Data.

Do první kategorie spadají především velcí poskytovatelé služeb z oblasti výpočetní techniky a internetu jako je například Microsoft, Amazon nebo Google. Microsoft nabízí službu HDInsight[17] zřízenou na platformě Microsoft Azure, Amazon zase Elastic Compute Cloud[18] (EC2), Simple Storage Service[19] (S3) nebo zpracovávající nástroj Elastic MapReduce[20] (EMR) a Google služby jako Cloud Dataproc[21], Cloud Storage[22] či BigQuery[23]. Všechny tyto nástroje umožňují více či méně spravovat a integrovat Hadoop, Spark a případně i jiné systémy a frameworky z Hadoop ekosystému, nebo nabízejí svá vlastní řešení a postupy, jak s Big Data pracovat a jak z nich získávat informace. Co se však týká samotné vizualizace, je jejich nabídka výrazně omezená, neboť se na ni tyto služby nezaměřují. Typicky je tak uživatelé využívají především jako úložiště, pro zpracování dat nebo k rychlému řešení analýz. Pro následnou vizualizaci jsou však již nuceni hledat jiná řešení.

Při využití distribuovaného systému a úložišť v cloudu je také výhodná existence přímého propojení této služby a nástroje, který se stará o samotné vykreslení. Tento způsob výrazným způsobem zjednodušuje práci, jelikož není potřeba manuálně stahovat a importovat data. Ačkoliv takových řešení není mnoho, existují výjimky. Příkladem je v předchozí podkapitole zmíněný GeoTrellis disponující možností spolupracovat s Amazon S3 nebo vizualizační nástroje Redash či Tableau umožňující přímé spojení s Google BigQuery.

V případě druhé možnosti, tedy pokud si potřebujeme nechat zpracovat celý proces vizualizace prostorových Big Data, lze například využít služby MapLarge[24] nebo MapD[25]. MapLarge představuje klasické proprietární řešení, kdy proces vykreslení probíhá na straně vlastního cloudu a k uživatelovi je přenesena pouze vizualizace ve formě mapových dlaždic jako překryvná vrstva nad určitým mapovým základem (například základní mapou OpenStreetMap). Tato služba nabízí mnoho možností vizualizace od bodového vykreslení dat přes heatmapu až po nejrůznější vzdálenostní analýzy. Jelikož se samozřejmě jedná o placenou službu, vše záleží pouze na zákazníkovi a jeho požadavcích.

[17] https://azure.microsoft.com/cs-cz/services/hdinsight/
[18] https://aws.amazon.com/ec2/
[19] https://aws.amazon.com/s3/
[20] https://aws.amazon.com/emr/
[21] https://cloud.google.com/dataproc/
[22] https://cloud.google.com/storage/
[23] https://cloud.google.com/bigquery/
[24] MapLarge. Dotupné z: http://www.maplarge.com/
[25] MapD. Dostupné z: https://www.mapd.com/

Naprosto odlišný a poměrně inovativní přístup poté volí společnost MapD, která vznikla teprve v druhé polovině roku 2013. Ta ke zpracování a vizualizaci prostorových Big Data nevyužívá počítačové clustery, ale více grafických karet. Místo mnoha strojů v distribuovaném systému a paralelního zpracování za pomoci více centrálních procesorů (CPU) tedy využívá pouze jeden počítač a paralelní zpracování za pomoci více grafických procesorů (GPU). Tyto čipy byly původně používány především k manipulaci s počítačovou grafikou a pro zpracování obrazu, nicméně v poslední době slouží i k jiným výpočtům, například v analytice, různých vědeckých aplikacích nebo obecně v oblasti zpracování dat. V některých případech může být vyřešení dotazu až stokrát rychlejší než u databází založených na CPU (Morgan 2016).

MapD nabízí dva hlavní produkty a to MapD Core, což je relační databázový systém umožňující díky zmíněnému přístupu dotazování biliónů řádků v milisekundách za využití standardního SQL, a poté MapD Immerse, vizualizační vrstvu, která dokáže se stejnou rychlostí data vykreslit. Služby MapD jsou tedy ideální především pro vytváření interaktivních a dynamických vizualizací, kde je potřeba rychle reagovat na změny a ihned je vykreslovat, popř. vytvářet vizualizace přímo v reálném čase.

Poskytovatelů služeb z oblasti vizualizace prostorových Big Data lze samozřejmě nalézt více, nicméně se ve všech případech jedná o proprietární a placená řešení. Především z tohoto důvodu jsou v této podkapitole představeni pouze dva zástupci zabývající se celým procesem vizualizace a stejně tak i jen několik poskytovatelů nabízející své clustery k pronájmu. Cloud a obecně služby třetích stran je sice nutné zmínit jako jednu z možností, jak k vizualizaci prostorových Big Data přistupovat, nicméně vzhledem k uzavřenosti použitých technologií a řešení není v zájmu této práce se daným tématem zabývat více dopodrobna.

## 4.3 Pokročilé vizualizační nástroje

Třetí možností, jak přistupovat k problému vizualizace prostorových Big Data, je využití pokročilých vizualizačních nástrojů představujících řešení v rámci jednoho počítače. Patří sem například technologie WebGL, která pracuje přímo s grafickou kartou, nebo speciální kombinace vizualizačních nástrojů umožňující zpracovávat větší objem dat, než je běžné. Nutno však podotknout, že u této kategorie jednak existuje horní hranice velikosti dat, která lze takto vizualizovat a jednak už je zde poněkud sporné, zda se jedná

o Big Data, nebo ne. Ačkoliv je objem dat často stále příliš velký, než aby šel zpracovávat běžnými nástroji v rozumném čase, hraje v otázce jejich označení pojmem Big Data roli především různorodost. Pro Big Data jsou totiž typická nestrukturovaná a semistrukturovaná data (viz kapitola 2), nicméně nástroje zde zmíněné pracují pouze s datovými sadami s pevnou strukturou nebo určitým předem definovaným datovým modelem.

Důvodem pro vytvoření této kategorie v rámci diplomové práce jsou především nejednoznačnosti spojené s pojmem Big Data, které mají za následek, že je některým vizualizačním nástrojům uživateli nebo autory přiřazena schopnost pracovat s Big Data, ačkoliv lze toto označení obecně považovat za nepřesné. Ve spojitosti s prostorovými daty může být tímto příkladem (Lurie 2013), (L'Astorina 2015), (Cherian 2013) nebo (Olshannikova et al. 2015) a jimi zmíněné javascriptové knihovny jako je především D3 či Leaflet. I přesto, že jsou zdroji těchto informací ve většině případů neodborné publikace, je i tak dobré na tyto nástroje upozornit. Nelze je samozřejmě srovnávat s distribuovanými systémy, které pracují s "pravými" Big Data definovanými často všemi vlastnostmi popsanými v druhé kapitole, nicméně vzhledem k jejich open-source licenci a existenci různých rozšíření je možné pomocí jejich vhodné kombinace vizualizovat větší objemy dat, než je běžné. Tento způsob navíc představuje pro uživatele dostupnější řešení, jelikož nejsou vyžadovány tak hluboké znalosti výpočetní techniky a programování jako u distribuovaných systémů. Stejně tak není potřeba vlastnit nebo si pronajímat více strojů. Naprostá většina uživatelů navíc potřebuje ve skutečnosti vizualizovat pouze větší objemy prostorových dat a ne "pravá" prostorová Big Data, která jsou, jak již bylo řečeno, většinou záležitostí pouze velkých společností a výzkumných center. V této podkapitole je tedy popsáno několik způsobů, jak lze vizualizovat větší objemy dat, přičemž otázka zda se jedná o Big Data nebo ne už záleží spíše na úhlu pohledu a zvolených kritériích hodnocení.

Především v této kategorii je poté nutné zdůraznit proces analýzy a filtrace dat. Tedy pochopení, co a jak předkládaná data představují, následované jejich transformací do strukturované podoby a odstraněním všech informací kromě těch, které jsou pro vizualizaci důležité (viz kapitola 3). Tyto kroky jsou totiž naprosto zásadní nejen pro přípravu dat pro konkrétní nástroj, ale především pro jejich redukci ještě před samotným vykreslením (Thompson et al. 2011). Odstraněním i jen jediného

nepotřebného atributu u všech prvků lze totiž snížit velikost objemnějších datových sad o znatelnou hodnotu. Odstranění všech nepotřebných informací poté může v krajním případě snížit velikost dat až na takovou hodnotu, že je lze vizualizovat běžnými nástroji. Zároveň je také žádoucí, a někdy i nutné, převést originální datovou sadu do jiného, ne tak objemově náročného, formátu. Příkladem může být konverze souborů založených na XML do JSON nebo CSV. V takových případech se však nesmí podcenit analýza a nastavení převodu především s ohledem na strukturovanou hierarchii původního formátu. Všechny tyto operace jsou důležité hlavně kvůli existenci horní hranice objemu dat, který lze nástroji z této kategorie zpracovat. Snahou je tedy snížit objem dat na co nejnižší úroveň, což kromě zmíněné možnosti vizualizace danými nástroji přináší i rychlejší práci a manipulaci se samotnými daty. Pro srovnání je poté vhodné uvést, že u distribuovaných systémů popisovaných v podkapitole 4.1 není tato potřeba tak významná, jelikož u nich nehraje velikost dat tak důležitou úlohu. Často je u nich naopak snahou ukládat surová data a až následnou analýzou z nich získávat potřebné informace či podklad pro samotnou vizualizaci.

První představenou technologií umožňující vizualizaci větších objemů dat s minimální odezvou je WebGL. Jedná se o javascriptové API pro akcelerované vykreslování grafiky za pomoci GPU v rámci kompatibilních webových prohlížečů bez nutnosti instalace zásuvných modulů (Parisi 2014). To znamená, že je kompletně integrováno do standardů webového prohlížeče, který umožňuje WebGL kreslit přímo do HTML elementu canvas. WebGL programy se skládají z obslužného kódu javascriptu a kódu tzv. shaderu. Ten je psán v jazyce GLSL a ovladač grafické karty ho kompiluje do kódu, který je vykonáván přímo na GPU (Danchilla 2012). Hlavní výhodou tohoto rozhraní je bezplatnost, multiplatformnost, široká podpora u desktopových i mobilních prohlížečů a samozřejmě mnohem rychlejší vykreslování grafiky než u běžného obsahu HTML canvas. To je dáno využitím paralelního zpracování na tisíci jádrech grafického procesoru.

Jednou možností je tedy využít přímo samotné API WebGL. U toho je však daní za vysokou flexibilitu a výkon poměrně složitější použití, neboť je nutné znát kromě javascriptu ještě jazyk GLSL a samotnou specifikaci WebGL. Pro uživatele je tedy výhodnější a především jednodušší využít nástroje, které jsou na této technologii založené. V souvislosti s prostorovými daty může být tímto příkladem javascriptová knihovna

WebGLayer, která byla rovněž vytvořena na ZČU. Nabízí možnost vykreslení všech bodů jako teček, vytvoření heatmapy i s interaktivní kontrolou parametrů a další užitečné funkce pro analýzu dat a jejich atributů. V příspěvku spojeném s touto knihovnou (Ježek et al. 2017) byla poté vytvořena demonstrativní aplikace zobrazující dopravní nehody ve Velké Británii. V této aplikaci byla analyzována rychlost vykreslení heatmapy pro různě velké datové sady zmíněnou knihovnou ve srovnání s ArcGIS Online, Google Maps Javascript API - Heatmap Layer a zásuvným modulem "Leaflet.heat" do knihovny Leaflet. Výsledky testu ukázaly, že pouze knihovna WebGLayer dokáže díky WebGL vykreslit heatmapu z více jak milionu prvků v čase, který umožňuje interaktivní práci s aplikací (tj. maximálně v řádu stovek milisekund). Ostatní řešení nebyla vůbec schopna větší datové sady vykreslit nebo potřebovala delší čas a to až v řádu desítek sekund.

WebGL je tedy vhodné pro webové aplikace s důrazem na rychlé vykreslení a následné překreslování vizualizace například na základě uživatelem měněných parametrů. Poradí si i s větším objemem dat, nicméně i zde existuje jeho horní hranice. Ta je daná více faktory jako je například počet dalších atributů u jednotlivých bodů či technické paramery grafické karty. Nevýhodou použití je poté fakt, že jednotlivé prvky nakreslené na elementu canvas nejsou součástí modelu DOM webové stránky, a tudíž k nim nelze jednoduše programově přistupovat. Kromě WebGLayer poté samozřejmě existují i další knihovny a nástroje pro prostorová data využívající WebGL. Příkladem může být Mapbox GL JS, knihovny pro zobrazení dat na sféře jako je Cesium nebo WebGL Eart, či Vizicities a OSM Buildings pro tvorbu map s 3D budovami. Dále lze WebGL využít i v kombinaci s již zmíněnými distribuovanými systémy. Tvorbou algoritmů pro vytvoření heatmapy spojením nástrojů Hadoop, HBase, Spark a právě WebGL je tématem například v (Perrot et al. 2015).

Další možností vizualizace objemnějších datových sad je využití javascriptových nástrojů pro redukci zobrazovaných dat, které jsou založeny na knihovně Leaflet. S ohledem na rychlost a maximální velikost zpracovávaného souboru patří mezi ty nejvýkonnější Supercluster a PruneCluster. Knihovnu Supercluster vytvořil tvůrce Leafletu, Vladimir Agafonkin, a její síla spočívá v aplikaci tzv. hierarchického nenasytného shlukování (hierarchical greedy clustering) rozšířeného o prostorový index. Díky tomuto přístupu zvládne knihovna pracovat i s 6 miliony bodů bez známky větší latence (Agafonkin 2016). Její nevýhodou je však časově náročnější počáteční inicializace.

Pro datové sady obsahujících do půl milionu prvků je délka tohoto úvodního výpočtu zanedbatelná, u větších datových sad poté autor poukazuje na možnost využít tzv. Web Worker. Ten obecně představuje řešení, jak spustit více vláken najednou, i přesto, že javascript sám o sobě vícevláknový není (Green 2012). V tomto konkrétním případě lze tak zpracování výpočtu vykonávat mimo hlavní vlákno, tedy na pozadí nezávisle na ostatních skriptech, čímž nedojde k zamrznutí prohlížeče nebo k blokaci vykreslení samotného mapového podkladu. Po úvodním zobrazení shluků jsou dotazy spojené s posuny a zoomováním obrazovky již otázkou maximálně stovek milisekund.

Druhým nástrojem pro vytváření shluků založeným na Leafletu a pracujícím s větším množstvím dat je PruneCluster. Jeho hlavní výhodou je možnost aktualizace shluků v reálném čase, takže může být použit například pro zobrazování velkého množství pohybujících se objektů. Další předností je podpora kategorizace v podobě možnosti barevného rozlišení zastoupení prvků dle jednotlivých kategorií v daném shluku. Nevýhodou je poté stejně jako u Superclusteru časově náročnější počáteční inicializace. Pro milion prvků, z nichž je polovina statická a polovina se pohybující, trvá úvodní výpočet více než jednu sekundu, následné posuny a zoomování obrazovky je už poté opět otázkou maximálně stovek milisekund.

Supercluster a PruneCluster tak představují efektivní řešení vizualizace většího množství dat především s ohledem na výslednou přehlednost a uživatelskou přívětivost aplikace, která je dána seskupováním obrovského množství bodových prvků do přiměřeného počtu shluků. Obě knihovny navíc umožňují přistupovat k jednotlivým bodům jako k objektům, což umožňuje například snadné zobrazení dalších atributů po kliknutí na daný bod. Společnou výhodou je také open-source licence, v případě Superclusteru ISC, u PruneClusteru MIT.

Další možností, jak vizualizovat větší objemy dat, je využít program Datashader v kombinaci s aplikací Jupyter Notebook. Ta představuje interaktivní webové prostředí, které umožňuje zpracovávat kód ve více než 40 programovacích jazycích. Její hlavní výhodou je možnost spouštět pouze krátké úseky kódu v uživatelem definovaných buňkách, takže lze snadno získávat jejich výstupy a mezivýpočty (viz obr. 6). Při ladění celého programu pak není potřeba ho vždy spouštět jako celek. Kromě toho mohou dokumenty vytvořené v této aplikaci obsahovat i rovnice, vizualizace, text nebo i videa.

In [3]:
```
import datashader as ds
import datashader.transfer_functions as tf
import pandas as pd
df = pd.read_hdf('SPOI.hdf5', '/coordinates/csv', columns=['latitude', 'longitude'])
#df = pd.read_csv('data_out.csv', usecols= ['latitude', 'longitude'])
df.tail()
```

Out[3]:

| | latitude | longitude |
|---|---|---|
| **27513803** | -20.503984 | 28.407441 |
| **27513804** | -20.502266 | 28.426533 |
| **27513805** | -20.517079 | 28.434906 |
| **27513806** | -17.925528 | 25.861529 |
| **27513807** | -18.935026 | 27.764414 |

In [4]:
```
World = (( -180,  180), (-90, 90))
x_range,y_range = World
plot_width  = int(2000)
#plot_height = int(plot_width*7.0/12)
plot_height = int(1000)
```

In [5]:
```
background = "black"
```

In [6]:
```
from functools import partial
from datashader.utils import export_image
from datashader.colors import colormap_select, Greys9
from colorcet import kbc
from IPython.core.display import HTML, display
export = partial(export_image, background = background, export_path="export")
cm = partial(colormap_select, reverse=(background!="black"))
display(HTML("<style>.container { width:100% !important; }</style>"))
```

Obr. 6 Ukázka prostředí Jupyter Notebook s importem nástroje Datashader a kusem kódu pro vizualizaci datové sady SPOI.

Programový balíček Datashader poté představuje nástroj pracující v rozhraní aplikace Jupyter Notebook. Umožňuje z kódu psaném v Pythonu vytvářet vizualizace extrémně velkých datových sad ve velice malém časovém intervalu i za použití běžného hardware na jediném počítači. Konkrétně jde o řády desítek milionů bodů vykreslených ve stovkách milisekund. Jeho zásadní nevýhodou je však možnost vytvářet pouze statické obrazy z pevné struktury neměnících se dat. Ideální je tedy pro reprezentaci prostorového rozložení velkého množství bodových prvků. Nicméně vzhledem ke skutečnosti, že nástroje založené na distribuovaných systémech nabízí ve většině případů také pouze statický výstup, představuje Datashader efektivní alternativu, jak tento typ vizualizace vytvořit i za použití jediného počítače. I díky tomu je v rámci práce využit pro tvorbu statické vizualizace datové sady SPOI. Jeho praktické použití i s kombinací dalších

programových balíčků je tedy předmětem kapitoly 5.2. Datashader i Jupyter Notebook jsou dostupné pod open-source licencí BSD.

Dalším představeným nástrojem je v rámci této podkapitoly desktopový software Tableau. Ten kromě placených produktů nabízí i verzi Tableau Public, která je k použití zdarma. Její hlavní předností je kromě vizualizace většího množství dat především jednoduché a intuitivní ovládání. Uživatel tak nepotřebuje znát žádný programovací jazyk ani nemusí disponovat pokročilými znalostmi z oblasti výpočetní techniky. Vše ovládá pouze v přehledném grafickém prostředí. Výstupem jsou interaktivní aplikace zobrazující uživatelovu datovou sadu vykreslenou na podkladě map OSM, se kterými lze pracovat (zoomovat, měnit oblast posunem apod.) stejně jako například s aplikací postavenou nad Leafletem. Jednotlivé body, případně vrstvy nebo oblasti, jsou navíc interaktivní a mohou po kliknutí zobrazovat další popisné informace. Samozřejmostí je také barevné rozlišení kategorií či klasifikace. Nevýhodou je poté především proprietární řešení, omezení počtu řádků (a tedy i bodů), se kterými může program pracovat, na maximálně 15 milionů a viditelná latence při práci s aplikací založené na větší datové sadě. Uživatel také musí pro získání instalačního balíčku registrovat svůj email. Další nevýhodou je omezení možnosti uložení vytvořené aplikace pouze na server Tableau. Až odtud je následně možné ji sdílet jako HTML objekt do svých vlastních stránek či aplikací. Více možností, výkonnější řešení a minimum omezení poté nabízejí další produkty této společnosti jako je Tableau Desktop či Server. Jelikož se však jedná o placené nástroje, lze je v rámci dělení možností vizualizace v této práci zařadit již spíše do druhé kategorie, tedy do využití služeb třetích stran.

V rámci této podkapitoly zmíněné nástroje představují naprosto rozdílné přístupy vizualizace. WebGLayer vykresluje bitmapové bodové prvky za pomoci WebGL, Supercluser a PruneCluster vytvářejí shluky bodů a na rozdíl od WebGLayer umožňují jednoduše programově přistupovat k jednotlivým bodům, Datashader v kombinaci s Jupyter Notebook zase dovoluje zpracovávat mnohonásobně větší datové sady, avšak pouze ve formě statických výstupů, a Tableau Public představuje zástupce desktopového programu nevyžadující na uživateli téměř žádné technické dovednosti či znalosti. Ani zde však není cílem práce dopodrobna srovnat a představit všechny existující programy a nástroje pro vizualizaci většího množství prostorových dat, u kterého je předmětem sporu, zda jde již o Big Data nebo ne. Vybráni tak byli pouze

zástupci reprezentující různé přístupy k vizualizaci. Snahou je tedy spíše poukázat na fakt, že na rozdíl od distribuovaných systémů existuje v této kategorii více možností, jak může být výsledná vizualizace zpracována. Je tady možné vybrat nástroj na základě toho, jak si uživatel výslednou vizualizaci představuje a jaké chce, aby měla vlastnosti. Tedy například zda bude interaktivní nebo statická, vektorová nebo bitmapová, jednoúrovňová nebo víceúrovňová, zda se využije některá forma redukce zobrazovaných dat či zda bude vykreslení probíhat u klienta nebo na serveru. U distribuovaných systémů tyto možnosti nejsou a uživatel se tak musí omezit pouze na několik základních forem vizualizace, které tyto nástroje nabízejí. Často tak může vytvořit pouze jednoúrovňový obraz a volí jen mezi variantou, zda se bude jednat o vykreslení bodových prvků nebo o heatmapu.

S vlastnostmi výsledné vizualizace také souvisí maximální hodnota velikosti dat, pro kterou lze nástroje z této podkapitoly ještě použít. Zde je nejprve nutné rozlišovat velikost vstupních dat a velikost souboru, se kterým přímo pracuje vizualizační nástroj. Jak již bylo řečeno, vstupní data jsou nejdříve podrobena analýze a filtraci. Předpokládejme tedy například, že je výstupem těchto operací soubor ve formátu CSV obsahující zeměpisnou šířku, délku a identifikátor. Není tedy důležité, zda byla vstupní data ve formátu XML o velikosti v řádu až deseti gigabajtů a obsahovala mnoho dalších atributových vlastností, nebo ve formátu JSON o stovkách megabajtů s pouze zmíněnými souřadnicemi a identifikátorem. Výstupem obou typů vstupních dat je po filtraci zmíněný soubor CSV pokaždé o stejné velikosti. Jedinou nevýhodou většího vstupního souboru je delší doba předzpracování, která však nehraje tak důležitou roli, neboť může být prováděna offline či odděleně od dalších kroků v procesu vizualizace, takže přímo neovlivňuje rychlost samotného vykreslení či načtení aplikace.

Mnohem významnější je tedy velikost předzpracovaného souboru, se kterým přímo pracuje vizualizační nástroj. A jelikož tento soubor obsahuje po filtraci ve většině případů již pouze hodnoty souřadnic a případně ještě několik málo dalších atributů, je pro nástroje z této podkapitoly důležitější spíše než velikost souboru počet prvků, které vykreslují. Respektive počet prvků, které zahrnují do výpočtů pro vytvoření vizualizace. Jeho přesná hodnota nelze u jednotlivých nástrojů přesně stanovit, jelikož rychlost vykreslení závisí například i na rychlosti internetového připojení nebo na technických parametrech počítače, nicméně je především determinována zmíněnými vlastnostmi výsledné vizualizace.

S ohledem na maximální počet prvků tak může být zásadní například rozhodnutí, zda při vykreslení bodově lokalizovaných prvků do interaktivní webové aplikace budeme chtít těmto bodům přiřadit i další popisné atributy, které se zobrazí po kliknutí na bod, nebo nám stačí prvky pouze zobrazit a budeme tedy zkoumat jen jejich prostorové rozložení. V prvním případě totiž musíme mít možnost k těmto bodům programově přistupovat, takže se jako nejsnazší cesta jeví integrovat data přímo do HTML objektů (DOM) a vykreslit je jako vektorovou grafiku (SVG). Každý bod tedy představuje jeden samostatný objekt, ke kterému lze snadno přistupovat například pomocí javascriptu. To umožňuje přiřadit k jednotlivým prvkům odpovídající atributová data, nebo jednoduše měnit jejich styl pomocí CSS. Na druhou stranu je však toto řešení náročné pro webový prohlížeč, který nezvládne zpracovávat tak velké množství SVG elementů. To vede ke snížení rychlosti celé aplikace nebo v krajním případě i k pádu či zamrznutí prohlížeče. V druhém případě, tedy pokud nepotřebujeme k jednotlivým prvkům přistupovat, ale pouze je zobrazit, je tak samozřejmě možné vykreslit a pracovat s mnohem více body. Místo SVG lze na tomto místě použít bitmapy, které vykresluje například HTML canvas za pomoci WebGL.

Pro praktickou ukázku možností a srovnání jednotlivých nástrojů představených v této podkapitole byla pomocí každého z nich vytvořena stejná demonstrativní vizualizace. Podkladem pro ni je již předzpracovaný soubor ve formátu CSV obsahující 1 200 000 prostorových prvků rozmístěných po celém světě. Použitá data byla extrahována z datové sady SPOI. Odkazy na výsledné aplikace pro praktické vyzkoušení online jsou součástí přílohy 2, zdrojové kódy se rovněž nacházejí na přiloženém DVD.

Výsledky testu ukazují (viz tab. 2), že si s vizualizací poradí všechny nástroje a u žádného z nich tedy nedojde k pádu programu či zamrznutí webového prohlížeče. Jak již bylo řečeno, doba prvního načtení a následná plynulost práce s aplikací závisí především na zmíněných vlastnostech aplikace, což tento test rovněž dokazuje. Co se týká interaktivních aplikací, vykazuje API WebGL, které neumí přistupovat k jednotlivým prvkům, nejmenší hodnoty latence. Tableau Public, které danou možnost nabízí, zase naopak největší. Nutno však podotknout, že aplikace Tableau Public je jako jediná spouštěna ze serverů společnosti Tableau a její chování tedy nemusí být determinováno pouze počtem bodů, ale i možnými technickými omezeními ze strany serveru

nebo rychlostí přenosu dat. PruneCluster[26] poté představuje kompromis mezi dvěma výše zmíněnými nástroji. Umožňuje totiž přistupovat k jednotlivým bodům a práce s ním je až na počáteční inicializaci rychlá. Nezobrazuje však zase intuitivně jejich prostorové rozložení, jelikož shlukuje polohově si blízké prvky a tím fakticky redukuje počty zobrazovaných prvků. Poslední nástroj, Datashader, poté jako jediný neumožňuje vytvoření interaktivní aplikace, nedovoluje tedy zoomovat podle úrovní nebo měnit oblast posunem. Na rozdíl od ostatních nástrojů, u kterých by se časy s rostoucím počtem prvků výrazně zvětšily a práce s aplikací by začínala být pro uživatele značně pomalá, je schopen během několika sekund zpracovat i mnohonásobně větší počet bodů. Po prvním spuštění kódu je navíc schopný i po změně parametrů a jeho úpravě vykreslit data pouze v řádu několika stovek milisekund.

Tab. 2 Srovnání a možnosti nástrojů pro vizualizaci většího množství dat na jednom počítači.

| **Název** | **Rychlost prvního načtení** | **Interaktivní** | **Možnost přistupovat k jednotlivým prvkům** | **Práce s aplikací** |
|---|---|---|---|---|
| WebGL | 2.2 s | ano | ne | < 200 ms |
| PruneCluster | 3.1 s | ano | ano | stovky ms pro velká měřítka<br>< 2 s pro malá měřítka |
| Datashader | 1.9 s | ne | ne | < 400 ms pro opakované spouštění i s úpravou parametrů |
| Tableau Public | 8.0 s | ano | ano | přibližně stejně jako první načtení 6-8 s |

Dále je na tomto místě ještě nutné zmínit další obecný přístup, jak vizualizovat velké množství prostorových dat za použití jediného počítače. Je jím rozdělení datové sady do mapových dlaždic a to buď v rastrové, nebo vektorové podobě. Výsledná aplikace poté

[26] Do popisovaného testu byl z dvojice nástrojů PruneCluster a Supercluster vybrán první zmíněný, jelikož dokáže pracovat i se souborem CSV. Supercluster umožňuje práci pouze s formátem GeoJSON, jehož velikost by i se stejnými daty byla výrazně větší, což by mělo negativní dopad na výsledné srovnání časů načtení jednotlivých aplikací.

nepracuje přímo s objemnými zdrojovými soubory, ale načítá již pouze předpřipravené dlaždice z aktuálního mapového pohledu a úrovně přiblížení. Jejich vytvoření se však obecně skládá z několika kroků a lze navíc využít různou kombinaci nástrojů.

V případě rastrového přístupu jsou výstupem čtvercové obrazy nejčastěji o velikosti 256 x 256 pixelů. Ty přitom mohou být vykreslovány na serveru podle aktuální potřeby aplikací nebo jsou již pro celé dané území předpřipraveny offline. V obou případech jsou však ke klientovi posílány už jen výsledné rastry, čímž je umožněno prezentovat velké množství dat bez ztráty na výkonu aplikací a zároveň výrazně minimalizovat potřebu přenášení velkých dat po síti. Zásadní nevýhodou je však nemožnost tyto dlaždice dále měnit či přistupovat k jednotlivým prvkům. Pro vykreslení dat lze použít například open-source nástroj Mapnik. Ten je navržen pro zpracování kompletních mapových podkladů a kromě možnosti spuštění na serveru je i základem mnoha desktopových klientských aplikací, jako je třeba TileMill, které se rovněž zabývají tvorbou mapových dlaždic. Z uživatelského pohledu je však velice složité ho nastavit a navíc je ovládán pouze přes příkazovou řádku. I tak je ale díky široké škále možností velice populární a je využit například pro vykreslování mapových podkladů OpenStreetMap[27], CartoDB[28] či Stamen[29]. Jako alternativu lze poté zmínit software Maperitive, který je dostupný jako snadno ovladatelná desktopová klientská aplikace rovněž generující kompletní mapové podklady. Její nevýhodou je však zase naopak nemožnost pracovat s příliš velkými soubory. Pro všechny nástroje a aplikace sloužící k tvorbě rastrových mapových dlaždic je poté společnou vlastností podpora nějakého druhu stylů. Obecně jde o dokument s přesně definovanou syntaxí, který stanovuje pravidla určující co, kdy a jak vykreslit.

Výše zmíněný přístup by tedy zvládl zobrazit a prezentovat celou datovou sadu SPOI, jejíž vizualizace je jedním z cílů této práce. Navíc by byl tento postup velice usnadněn a to díky skutečnosti, že data obsahují pouze jediný datový typ a to bod. Definice stylů by tak případně obsahovala pouze barvu tečky reprezentující daný prvek, její velikost v jednotlivých úrovních a nastavení transparentního pozadí. Pro tento konkrétní případ by šel navíc místo poměrně náročného toolkitu Mapnik využít nástroj Datamaps. Ten je vytvořen speciálně pro dynamické generování mapových dlaždic z rozsáhlých seznamů souřadnic bodů nebo linií včetně základního vizuálního nastavení. Jeho výhodou je kromě

[27] https://www.openstreetmap.org/
[28] https://carto.com/location-data-services/basemaps/
[29] http://maps.stamen.com/

snadnějšího použití i rychlost, jelikož zdrojová data indexuje a kóduje do čtyřstromu podle Mercartorova zobrazení. Výsledné mapové dlaždice by šlo poté připojit do aplikace jako překryvnou vrstvu nad již existujícími mapovými podklady jako je například OpenStreetMap. Vzhledem k celosvětovému pokrytí bodů SPOI je však zásadní nevýhodou nutnost vytvořit mapové dlaždice pro všechny úrovně pro celý svět. Jednak by tato operace trvala velice dlouhou dobu a navíc by vygenerované soubory zabíraly příliš diskového prostoru a to až v řádu tisíců gigabajtů. Tento problém by však šel částečně obejít kreslením dlaždic až na serveru podle aktuálních požadavků. Další a zcela stěžejní nevýhoda poté vychází přímo z podstaty rastrové reprezentace mapových dlaždic. Jednou vykreslené body už jsou totiž neměnné a nelze k nim přistupovat. Není tedy možné zobrazit ani procházet další atributy, které jsou s nimi spojené. Tato možnost je však z pohledu prezentace datové sady SPOI zásadní. Řešení pomocí rastrových mapových dlaždic tedy proto není využito a to i přesto, že dokáže nejen zobrazit všechny body, ale lze pomocí něj vytvořit i aplikaci umožňující plynulý průzkum dat s minimální latencí.

Druhou možností je poté využít mapové dlaždice ve vektorové formě. Jejich hlavní výhodou je především rychlost a flexibilita. Místo rastrů jsou totiž posílány ke klientovi vektory, které jsou vykreslovány až samotnou aplikací na základě jejích požadavků. To znamená, že jsou u klienta dostupné bez ztráty všechny informace o datech včetně jejich atributů a je také možné k nim snadno přistupovat. Dále lze jednoduše měnit vizuální parametry vykreslení jako například použité barvy nebo nastavení symbolů, či dokonce vypínat jednotlivé vrstvy. Prakticky tak může více nezávislých aplikací vykreslit stejné vektorové mapové dlaždice naprosto rozdílným stylem nebo ho dynamicky měnit například podle požadavků uživatele. Pro mnoho způsobů vykreslení či více aplikací tedy stačí vytvořit pouze jedinou službu distribuující vektorové mapové dlaždice v rámci internetu. Jelikož je navíc taková dlaždice tvořena pouze geometrickými a popisnými informace, jsou kladeny ještě nižší nároky na síťový provoz a na velikost datového úložiště než u rastrů.

Konkrétním využitelným nástrojem pro generování vektorových dlaždic je poté například geojson-vt. Ten je dokáže vytvářet za běhu (tzv. “on the fly”) i bez použití serveru. Zvládne přitom pracovat s několika miliony bodů bez viditelné latence. Nevýhodou je pouze časově náročnější prvotní inicializace. Pokročilejším řešením je poté nástroj Tippecanoe speciálně navržený pro převod velkých kolekcí prostorových dat

do vektorových dlaždic. Ty jsou následně distribuovány pomocí serveru, který vyřizuje požadavky klientských aplikací na přenos konkrétních dlaždic. Tímto způsobem lze tedy úplně eliminovat výše zmíněný problém s latencí při prvotním spuštění a využít tak všech výhod vektorové reprezentace. Je tak možné vytvořit vizualizaci umožňující nejen plynule prozkoumávat i desítky miliony prvků, ale i přistupovat k jejich atributům. Právě z tohoto důvodu je nástroj Tippecanoe a obecně řešení s vektorovými mapovými dlaždicemi využito pro interaktivní vizualizaci datové sady SPOI, která je předmětem kapitoly 5.3.

# 5. Vizualizace datové sady SPOI

Druhá část diplomové práce se věnuje vizualizaci datové sady Smart Points of Interest. Z důvodu její velikosti a obsahu velkého počtu bodů ji není možné zpracovat a vykreslit běžnými nástroji a je tedy nutné použít pokročilejší techniky nebo přímo nástroje pro prostorová Big Data. Tato kapitola se tedy zabývá analýzou zmíněné datové sady a popisuje její následnou praktickou vizualizaci.

## 5.1 Analýza datové sady

Název Smart Points of Interest (SPOI) představuje bezešvou datovou sadu obsahující body zájmu se zaměřením na cestovní ruch a s ním příbuzná odvětví. Prezentuje tedy místa, která jsou z tohoto pohledu nějakým způsobem zajímavá nebo užitečná. Jedná se o otevřený a volně přístupný zdroj, který mohou ostatní uživatelé využívat i pro jiné aplikace či služby nebo jen libovolně stahovat či vyhledávat potřebné body. Datová sada pokrývá území celého světa a její velká přednost spočívá především v implementaci propojených dat (Linked Data) a využití standardizovaných datových typů a jejich vlastností (Čerba a Mildorf 2017). Je vyvíjena v rámci mezinárodního projektu SDI4Apps - Uptake of Open Geographic Information Through Innovative Services Based on Linked Data, který je financován Evropskou Unií, koordinován Západočeskou univerzitou v Plzni a je zaměřen na využití otevřených geografických informací pomocí inovativních služeb založených na Linked Data. Po skončení tohoto projektu bude datová sada dále vyvíjena a spravována pod projektem Peregrinus Silva Bohemica - Multimediální a digitální turistický průvodce pro přeshraniční historické cesty v Bavorském lese a na Šumavě.

Zdrojem dat SPOI jsou jak vybraná globální data extrahována například z projektů OpenStreetMap či GeoNames, tak i místní data volně dostupná z internetu či poskytována partnery SDI4Apps (region Pošumaví, data ze Sicílie, historické památky v Římě aj.). Kromě toho je všem uživatelům umožněno přidávat nové body nebo upravovat ty stávající, díky čemuž mohou i dobrovolníci datovou sadu SPOI libovolně rozšiřovat a vylepšovat. Všechna data z velice heterogenních zdrojů jsou následně v rámci procesu harmonizace transformována do otevřeného a flexibilního datového modelu. Ten je neustále rozšiřován

a kromě povinných atributů jako jsou například souřadnice, identifikátor či klasifikace, obsahuje i další vlastnosti, které mohou být pro turistiku užitečné (adresa, telefon, webové stránky aj.). Pro popis dat jsou přitom využity již existující a běžně používané slovníky jako je Dublin Core, GeoSPARQL, Web Ontology Language (OWL) nebo Friend of a Friend (FOAF). Dále jsou obsahem SPOI i topologické vazby na příslušné státy v rámci databází GeoNames či DBpedia, vazby na identické prvky v projektech Wikidata, LinkedGeoData, DBpedia a odkazy na další nestrukturované dokumenty jako jsou například fotky nebo mapy. Detailnější informace o datové sadě, použitých standardech a metodologii jsou poté k dispozici na oficiálních webových stránkách SPOI[30] a v mnoha odborných příspěvcích či prezentacích s ní spojených (Čerba a Mildorf 2017), (Čerba 2017), (Čerba et al. 2016a), (Čerba et al. 2016b).

```
<rdf:Description rdf:about="http://www.sdi4apps.eu/poi/#OSM_26864258">
    <rdfs:label>Pica d'Estats</rdfs:label>
    <geos:asWKT rdf:datatype="http://www.openlinksw.com/schemas/virtrdf#Geometry">
    POINT(1.3978986 42.666952)</geos:asWKT>
    <poi:class rdf:resource="http://gis.zcu.cz/SPOI/Ontology#peak"/>
    <poi:class rdf:resource="http://gis.zcu.cz/SPOI/Ontology#natural_feature"/>
    <owl:sameAs rdf:resource="http://linkedgeodata.org/triplify/node26864258"/>
    <skos:exactMatch rdf:resource="http://linkedgeodata.org/triplify/node26864258"/>
    <geos:sfWithin rdf:resource="http://www.geonames.org/3041565"/>
    <geos:sfWithin rdf:resource="http://dbpedia.org/resource/Andorra"/>
    <dc:identifier rdf:resource="http://www.sdi4apps.eu/poi/#OSM_26864258"/>
    <dc:publisher>SPOI (http://sdi4apps.eu/spoi)</dc:publisher>
    <dc:title>Pica d'Estats</dc:title>
    <dc:rights rdf:resource="http://opendatacommons.org/licenses/odbl/1.0/"/>
    <dc:source rdf:resource="https://www.openstreetmap.org"/>
    <dcterms:created rdf:datatype="http://www.w3.org/2001/XMLSchema#date">
    2016-11-21</dcterms:created>
  </rdf:Description>
```

Obr. 7 Ukázka bodu zájmu ve formátu RDF v rámci datové sady SPOI.

Z pohledu vizualizace jsou důležité především technické specifikace datové sady SPOI. Ta obsahuje celkem 27 513 808 bodů (stav k lednu 2017) a jedná se tak o největší otevřenou datovou bázi na světě publikovanou na principu Linked Data (Čerba 2017). Výstupem procesu harmonizace a transformace zdrojových dat do používaného modelu je celkem 553 souborů ve formátu RDF rozdělených ve většině případů podle státu a zdroje

[30] http://sdi4apps.eu/spoi/

dat. Jejich celková velikost je více než 30 GB. Konkrétní ukázka kódu jednoho bodu je zachycena na obrázku č. 7. Pro uživatele však nejsou zmíněné soubory dostupné, neboť je datová sada nahrávána do databázového enginu OpenLink Virtuoso a publikována přes SPARQL endpoint, který umožňuje standardizované dotazování a stahování pomocí sémantického dotazovacího jazyka SPARQL.

S ohledem na téma diplomové práce je také důležité zhodnocení datové sady z pohledu Big Data. Ta je autory SPOI takto označena (Čerba a Mildorf 2017) a s odkazem na obecnou definici Big Data, která je prezentuje jako data, se kterými neumíme pracovat běžnými nástroji a v rozumném čase (Snijders et al. 2012), lze s tímto tvrzením souhlasit. Vizualizovat více jak 27 milionů bodů není běžnými nástroji ani v delším časovém horizontu možné a stejně tak zpracování dat o objemu více jak 30 GB je především s ohledem na využití paměti taktéž pro většinu nástrojů neřešitelné.

Analýzou SPOI z pohledu vlastností Big Data popisovaných v druhé kapitole už však toto označení pro popisovanou datovou sadu tak jednoznačné není. Její objem sice neumožňuje snadnou manipulaci s daty, nicméně Big Data jsou obecně spojována s mnohem větší velikostí a to až v řádu stovek gigabajtů, terabajtů či petabajtů, která velice často ani nelze uložit na jediném počítači. Objem SPOI je navíc rozdělen do zmíněných 553 souborů, přičemž největší z nich dosahuje velikosti 3,4 GB. Datová sada se však neustále zvětšuje a její objem tak může teoreticky růst do mnohem vyšších hodnot. Za rok 2016 byly například přidány 3 miliony bodů (Čerba 2017). Další ze základních vlastností, která charakterizuje Big data, je požadavek na rychlé zpracování dat, v krajním případě na jejich zpracování v reálném čase. Aktualizace SPOI však probíhá 4 krát ročně (Čerba 2017) a na rychlost tedy nejsou kladeny žádné speciální požadavky. V termínu aktualizace se tak nahrají všechna nová data, která byla do té doby transformována, a ta, která nestihla projít procesem harmonizace, se nahrají až v termínu dalším. Třetí vlastností ze základního „3V“ modelu charakterizujícího Big Data je poté různorodost. Zde je datová sada SPOI opět v rozporu s označením Big Data, jelikož ji netvoří nestrukturovaná nebo semistrukturovaná data. Jejím zdrojem je sice mnoho heterogenních dat, ale ta jsou transformována do předem definovaného datového modelu a pevné struktury v podobě formátu RDF. Jediným náznakem spadajícím pod Big Data je flexibilnost datového modelu, který svými změnami reaguje na aktuální potřeby a je tak často měněn nebo rozšiřován. Během roku 2016 v něm došlo ke dvanácti změnám (Čerba 2017).

Z pohledu zmiňovaných vlastností je tedy už označení datové sady SPOI jako Big Data poněkud sporné. Zároveň je však na tomto místě ale nutné podotknut, že s ohledem na vizualizaci, která vyžaduje speciální přístupy už i při zobrazení méně jak jednoho milionu prvků (viz kapitola 4.3), představuje SPOI díky obsahu 27 milionů bodů poměrně složitý problém, který lze z tohoto pohledu pod Big Data zařadit. Jak bylo totiž řečeno v druhé kapitole, pojem Big Data není pouze o datech, ale dá se na něj nahlížet obecně i jako na problém, který není řešitelný běžnými nástroji. Jinými slovy lze říci, že pokud bychom chtěli s touto datovou sadou pouze manipulovat nebo nad ní provádět analýzy, bylo by její označení jako Big Data sporné. Nicméně pokud je potřeba ji vizualizovat, vstupují v potaz i další argumenty jako na například požadavek na co nejrychlejší vykreslení dat. Z tohoto hlediska lze tedy problém vizualizace SPOI popsat jako problém Big Data. U samotné datové sady záleží poté už více na úhlu pohledu, zvolených kritériích hodnocení a je již otázkou diskuse, zda ji lze již označit jako Big Data či nikoliv.

Co se týká samotné vizualizace dat SPOI, existuje již mapový klient, který slouží uživatelům k prohlížení dat. Je vytvořen pomocí knihovny HSLayers-NG, která rozšiřuje OpenLayers, a kromě několika podkladových map (OpenStreetMap, OpenCycleMap[31] aj.) nabízí i možnost zobrazit či skrýt jednotlivé vrstvy bodů SPOI rozdělené podle deseti základních kategorií (například doprava, kultura a zábava či nakupování a služby) nebo podle několika vybraných podkategorií (restaurace, banky). Jednotlivé body jsou přitom zobrazovány jako obrázkové ikony, které po kliknutí zobrazují další informace a daném bodu zájmu. Kromě toho obsahuje mapový klient například i informace o počasí, možnost mapu sdílet nebo vytvářet vlastní výlety s automatickým routováním mezi vybranými body.

Z hlediska samotné vizualizace je poté u zmíněného mapového klienta důležité, že jsou data extrahována z databázového enginu Virtuoso pomocí SPARQL dotazu, který je generován přímo aplikací. Ten vybírá vždy potřebná data na základě prostorového dotazu a selekce jednotlivých bodů je tedy vymezena obdélníkem daným souřadnicemi aktuálního pohledu uživatelovo obrazovky. Nicméně mapový klient nedokáže zobrazit větší množství dat v menších měřítkách, což je dáno především dvěma hlavními důvody. První souvisí se službou Virtuoso, které automaticky ukončuje kvůli dlouhému času zpracování dotazy, jejichž výsledkem jsou objemná data. Nicméně i pokud by k tomuto

[31] https://www.opencyclemap.org/

přerušení nedocházelo, zobrazení dat v mapové aplikaci by trvalo příliš dlouho, neboť je nutné nejprve vyřešit prostorový dotaz na straně enginu Virtuoso, poté výsledek poslat po síti ke klientovi a zde ho následně zpracovat a zobrazit. Jelikož je však takový dotaz vytvářen při každém posunu v mapě či změně měřítka, nebylo by možné kvůli velké latenci s aplikací plynule pracovat. Druhým důvodem je poté velké množství dat, s kterými aplikace pracuje. Při načtení více bodů i ve větších měřítkách se totiž dostává do problémů webový prohlížeč, který nedokáže zpracovávat tak velké množství objektů. Jednotlivé ikony se poté načítají opět příliš dlouho nebo se při posunu na jiné místo nezobrazí už vůbec a je potřeba celou aplikaci ve webovém prohlížeči znovu načíst.

Výše popsané problémy a nemožnost zobrazit data pro menší měřítka jsou hlavním důvodem vizualizace datové sady SPOI v rámci této práce. Cílem je tedy pokusit se vytvořit aplikaci, která bude prezentovat všechna data a navíc uživatelsky přívětivým způsobem, tj. atraktivně a s co nejmenší latencí.

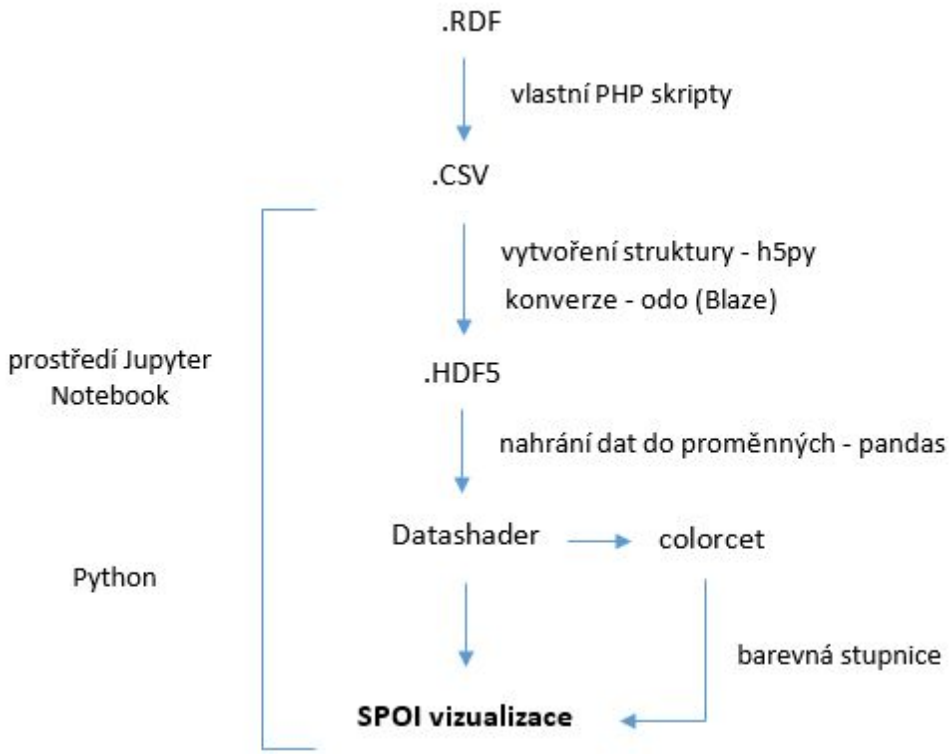
V rámci prezentace datové sady SPOI je ještě nutné zmínit již vytvořenou aplikaci[32] zabývající se vizualizací a analýzou jejích dat pro oblast Lotyšska. Je vytvořena autorem této diplomové práce v rámci projektu SDI4Apps a pro zobrazení dat (více jak 100 000) využívá knihovnu WebGLayer zmíněnou v kapitole 4.3. Aplikace tak pomocí technologie WebGL a podpory knihovny D3 umožňuje s minimální odezvou interaktivní průzkum a analýzu dat SPOI v podobě bodových znaků, nebo heatmapy i s možnou úpravou jejích parametrů. Navíc je nabízena možnost filtrace dat a to buď podle deseti hlavních kategorií nebo na základě prostorového vymezení pomocí uživatelem definovaného polygonu. Na rozdíl od výše zmíněného mapového klienta vytvořeného pomocí HSLayers-NG, nepracuje tato aplikace s databázovým enginem Virtuoso a SPARQL dotazy, ale přímo se zdrojovými RDF soubory.

## 5.2 Tvorba statické vizualizace

Pro potřeby prezentace Smart Points of Interest bylo rozhodnuto vytvořit nejprve statický obraz, který bude zobrazovat celou datovou sadu v měřítku zahrnující celý svět. Snahou je tedy vykreslit všech více jak 27 milionu bodů bez ohledu na kategorizaci a demonstrovat tak prostorové rozložení datové sady SPOI jako celku. Pro tvorbu mapy byl jako základní nástroj vybrán Datashader popisovaný v kapitole 4.3, který zvládne

[32] http://portal.sdi4apps.eu/spoi-webglayer/

na jednom počítači a v krátkém čase vykreslit i desítky miliony bodů. Schematicky znázorněný postup tvorby statické vizualizace je poté zachycen na obrázku 8.




Obr. 8 Schéma tvorby statické vizualizace SPOI.

Prvním stěžejním krokem v procesu vizualizace je předzpracování dat, konkrétně především jejich filtrace. Úkolem je tedy na tomto místě získat z popisu jednotlivých bodů ve formátu RDF pouze souřadnice, jelikož žádné jiné informace nejsou pro tuto konkrétní vizualizaci potřebné. Zároveň se eliminací všech nedůležitých informací výrazně sníží objem celé datové sady. Bude tedy vhodné uložit výsledek předzpracování do jediného souboru ve formátu CSV, čímž se výrazně ulehčí práce s datovou sadou v nástroji Datashader. Autorem této práce byl tedy vytvořen skript v programovacím jazyce PHP, který postupně načítá všech 553 zdrojových souborů SPOI a nalezené souřadnice vypisuje odděleně jako zeměpisnou šířku a délku do souboru CSV. Zároveň tento skript neběží jako jeden dlouhý proces, ale efektivně po každém zpracovaném souboru čistí paměť a uživateli průběžně na obrazovku vypisuje informace o stavu celé konverze. Ze strany uživatele tak není potřeba žádná interakce, pouze nastavení umístění a jmen vstupních a výstupního souboru. Nevýhodou skriptu je poté nutnost disponovat serverem a potřeba

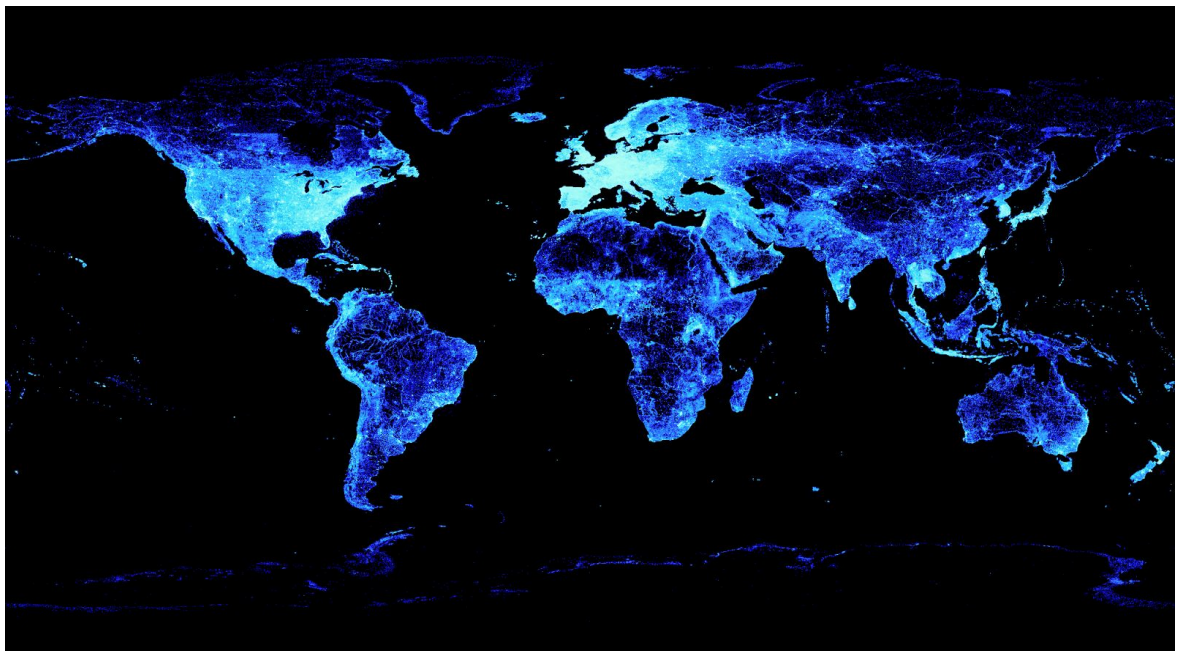
změnit v nastavení PHP maximální limit paměti a maximální dobu realizace jednoho skriptu na neomezenou hodnotu. Skript také může mít problémy na počítačích s horšími technickými parametry.

Na tomto místě se tak jeví jako ideální využít pro proces předzpracování dat právě distribuované systémy, které jsou přímo vytvářeny pro práci s velkoobjemnými soubory. Je nutné si však uvědomit, že nastavení a konfigurace takového systému je z pohledu uživatele poměrně složité, a pokud je daná úloha řešitelná na jednom počítači, měla by být řešena spíše tímto způsobem. V případě distribuovaného systému by uživatel musel navíc disponovat počítačovým clusterem a ne pouze jedním serverem. I za cenu delšího zpracování tak bylo rozhodnuto vytvořit skript běžící pouze na jednom počítači, který však po uživateli nevyžaduje žádné dovednosti, stačí pouze na serveru zadat umístění souboru a celá konverze již proběhne sama.

Další práce už poté probíhaly pouze v interaktivním webovém prostředí nástroje Jupyter Notebook a výhradně v programovacím jazyce Python. Prvním krokem byl převod předzpracovaného výstupního souboru CSV do formátu HDF5, jelikož byl jeho objem stále poměrně velký, než aby se s ním dalo jednoduše manipulovat. HDF5 představuje samopopisující hierarchický datový formát, který může v jednom souboru uchovávat tisíce datových sad různých datových typů (Collette 2013). Je tedy vhodný pro ukládání a organizaci Big Data a jeho hlavní výhodou je především možnost rychlého přístupu k datům. Pro vytvoření struktury tohoto souboru byl využit programový balíček h5py a ke konverzi dat mezi CSV a HDF5 balíček Odo z tzv. Blaze ekosystému, který slouží jako rozhraní pro práci s Big Data. Výsledkem je tedy soubor ve formátu HDF5 obsahující datovou skupinu “coordinates”, ve které jsou podobně jako sloupce v tabulce odděleny hodnoty zeměpisné šířky a délky pro jednotlivé body. Velikost tohoto souboru je navíc ještě o necelých 30 % menší než velikost souboru CSV.

Další kroky už poté nejsou spojeny s předzpracováním dat, ale již přímo s jejich reprezentací. Pro nahrání jednotlivých souřadnic bodů do proměnné v rámci programového prostředí byl využit nástroj pandas. Ten zajišťuje čtení dat ze souboru HDF5, jejich základní analýzu jako je počet a rozsah souřadnic, a následnou snadnou a především rychlou manipulaci s proměnnou, která tato data obsahuje. Dále bylo nutné nastavit parametry vizualizačního nástroje Datashader jako je například velikost plátna, vymezení zobrazovaného území nebo použité barvy. Pro ty byl navíc ještě integrován programový

balíček colorcet obsahující předem definované barevné palety. Ty zajišťují atraktivnější vzhled, neboť jsou vytvořeny s ohledem na lidské percepční schopnosti a na rozdíl od výchozích barev Datashaderu působí věrněji a zachycují více detailů. Kromě prostorového rozložení bodů SPOI tak výsledná vizualizace vyjadřuje díky modré barevné stupnici ještě i hustotu bodů v jednotlivých oblastech světa. Ukázka finálního vykreslení je zachycena na obrázku 9, na přiloženém DVD je poté k dispozici vizualizace v plném rozlišení 4000 x 2000 pixelů.



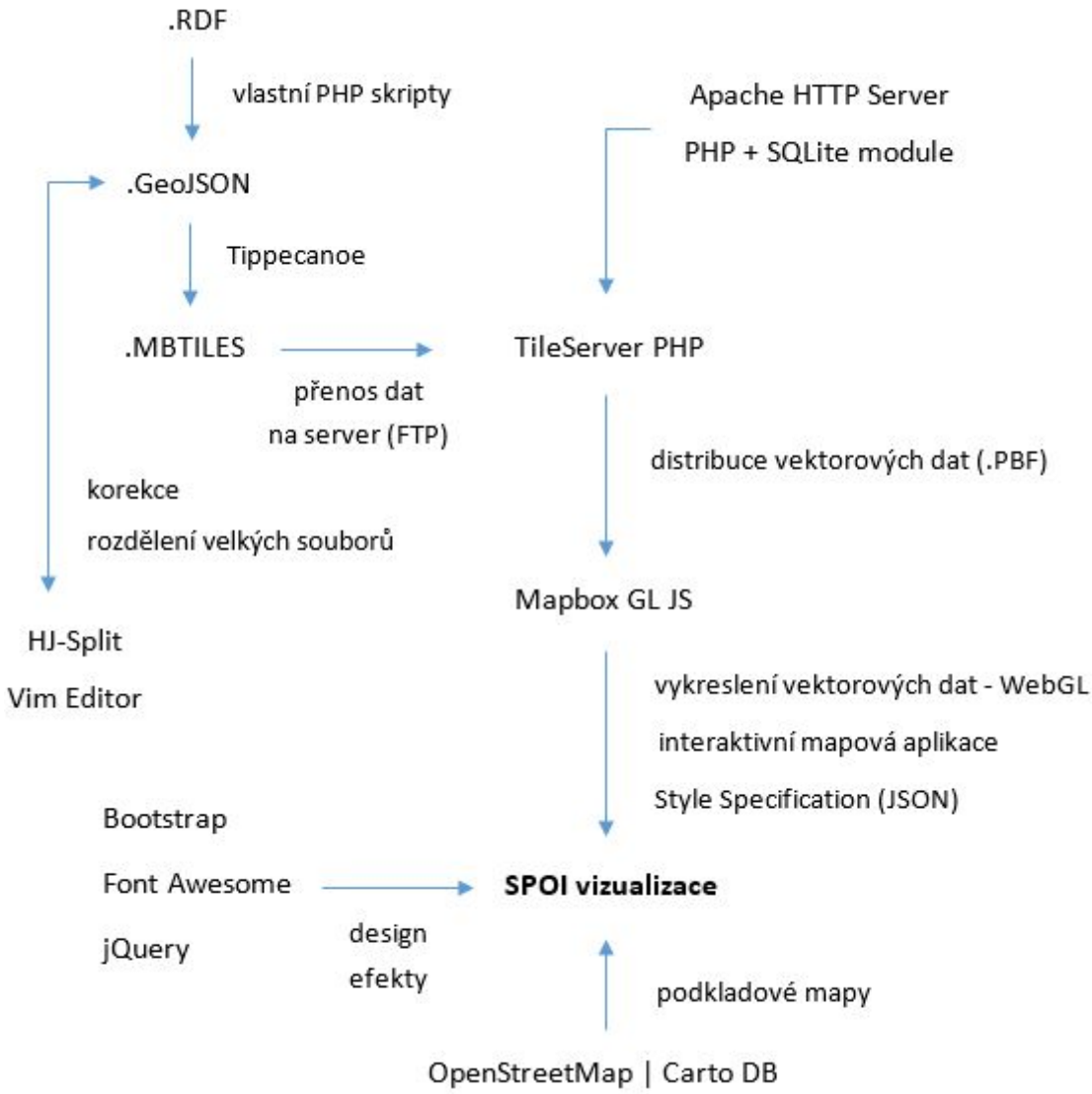
Obr. 9 Statická vizualizace datové sady SPOI s aplikací barevné stupnice.

## 5.3 Tvorba interaktivní aplikace

Hlavním cílem praktické části této práce je poté vytvořit interaktivní webovou aplikace, která bude prezentovat všechny body SPOI a navíc umožní zobrazit i jejich další popisné atributy. Pro tento úkol bylo vybráno řešení převádějící zdrojová data do vektorových mapových dlaždic (viz kapitola 4.3), která jsou dále přes vytvořený server přenášena ke klientovi a zde následně vykreslena do výsledné podoby. Celý postup je schematicky znázorněn na obrázku 10.

Prvním krokem je stejně jako u statické vizualizace proces předzpracování. Opět se zde vychází ze zdrojových souborů ve formátu RDF, který je vhodný pro sémantický popis dat, nicméně pro vizualizační nástroje již nikoliv. V jazyce PHP byly

tedy opět vytvořeny skripty transformující všechna data SPOI tentokrát do formátu GeoJSON. Konkrétně bylo vytvořeno deset těchto souborů reprezentující deset hlavních kategorií v datech, které budou tvořit vrstvy ve výsledné aplikaci (Car Service, Culture & Entertainment, Food And Drink, Lodging, Natural Feature, Other, Outdoor, Professional And Public, Shopping And Service, Transportation). Jelikož se však na rozdíl od předzpracování u statické vizualizace transformovaly všechny informace o bodech a ne jen jejich souřadnice, bylo nutné je také kontrolovat a případně i upravit. V komentářích či popisech u jednotlivých bodů se totiž často vyskytují uvozovky, znaky konce řádků či jiné znaky narušující strukturu formátu GeoJSON. Součástí vytvořených PHP skriptů je tedy kromě automatického rozdělení bodů do jednotlivých souborů podle kategorií i detekce a úprava chybných znaků ve všech atributech.




Obr. 10 Schéma tvorby interaktivní webové aplikace prezentující SPOI.

Z pohledu řešení vizualizace velkého množství bodů SPOI je poté naprosto zásadní open-source nástroj Tippecanoe psaný v C++. Ten převádí jednotlivé prvky z kolekce ve formátu GeoJSON do vektorových mapových dlaždic. Jeho velkou předností je především možnost transformovat miliony bodů, omezen je pouze velikostí zpracovávaného souboru na několik GB. Právě z tohoto důvodu musely být některé soubory s kategoriemi (Natural Feature, Other, Professional And Public a Transportation) nejprve rozděleny na více částí. K tomu byl využit nástroj HJ-Split, který je stejně jako Vim Editor, použitý pro úpravu hlavičky a konce rozdělených částí, navržen pro práci s velkoobjemnými soubory.

Výstupem nástroje Tippecanoe je poté formát s otevřenou specifikací MBTiles. Technicky je jedná o SQLite databázi, která umožňuje uložení vektorových (ale i rastrových) mapových dlaždic v jediném souboru. Pro každou kategorii je tedy vytvořen jeden takový soubor, přičemž pro kategorie, které byly rozděleny na více částí, je těchto souborů o to více.

V nástroji Tippecanoe je již také potřeba nastavit generalizaci jednotlivých prvků tak, aby se v malých měřítkách nezobrazovalo všech 27 milionů bodů, což by vedlo k naprosté nepřehlednosti a nečitelnosti výsledné vizualizace. Pro každou vrstvu se tedy musela s ohledem na počet bodů zvolit základní úroveň, ve které se poprvé zobrazí všechna data SPOI. Od ní se poté s každou úrovní směrem k menším měřítkům vždy určité procento bodů vynechá v závislosti na nastavené míře generalizace. Zde je nutné však zdůraznit, že se tímto krokem nenastavují parametry vykreslení jednotlivých bodů, ale určuje se přímo obsah samotné vektorové reprezentace dat. To znamená, že vynechané body v jednotlivých úrovních vůbec nevstupují do dalšího zpracování a následného vykreslení, čímž se výrazně snižuje objem výstupního souboru ve formátu MBTiles. Při přípravě rastrových mapových dlaždic se totiž typicky používají veškerá data a pouze se určuje, kdy a jak se daný prvek vykreslí. Navíc je jejich tvorba oddělena od samotné aplikace a k uživateli jsou tedy přenášeny už jen výsledné obrazové dlaždice nejčastěji ve formátu PNG. U vektorového základu je nastavení vykreslení, tj. například velikost tečky reprezentující jeden bod v různých úrovních zoomu, implementováno až v kódu samotné aplikace a jednotlivé body se tedy ve většině případů vykreslují až na straně klienta.

Vzhledem k nerovnoměrnému rozložení bodů SPOI je však velice složité nastavit jak základní úroveň, tak i míru generalizace. Velká část dat je totiž vázána k Evropě, kde existují oblasti, zejména velká města, s velice hustou koncentrací bodů. Druhým extrémem je poté například Afrika, kde se naopak nacházejí velké oblasti, ve kterých není žádný nebo pouze několik bodů. V těchto místech je tedy snahou zobrazit data co nejdříve (tj. v co nejmenším měřítku), aby uživatel nemusel dlouho posouvat mapou ve velkém přiblížení, než se dostane k dalšímu bodu. U oblastí s jejich vysokou koncentrací je však naopak výhodné zobrazit všechna data až v měřítkách větších, jelikož se jednotlivé tečky překrývají a nelze je tudíž rozlišit či snadno kliknou přímo na jeden požadovaný bod. Jako kompromis byla pro výslednou aplikaci stanovena úroveň přiblížení 13 jako ta, kdy se směrem od malého měřítka (úroveň 0 pro zobrazení celého světa) poprvé zobrazí úplně všechny body pro všechny kategorie. Z pohledu metody teček jako jedné z metod tematické kartografie tak tato úroveň představuje hranici, od které jsou směrem k větším měřítkům všechny tečky umístěny lokalizovaným způsobem, tj. přesně na daných souřadnicích, zatímco směrem k menším měřítkům je jejich geometrie mírně upravena tak, aby se tečky co nejméně překrývaly. I přesto však v oblastech s vysokou koncentrací bodů k překryvům dochází, především u vrstvy Natural Feature, která tvoří přibližně 30 % všech dat SPOI. Zde je nutné také upřesnit, že výše zmíněné nastavení je vytvářeno vzhledem k jednotlivým vrstvám zvlášť, tudíž při zobrazení všech kategorií najednou bude k překrývání docházet téměř vždy.

V nástroji Tippecanoe bylo dále ještě nutné upravit výchozí nastavení, které stanovuje limit jedné vektorové dlaždice na maximální velikost 500 000 bajtů a na maximální obsah 200 000 prvků. Tato omezení vycházejí z požadavků služeb, které umožňují připojit mapové dlaždice do koncových aplikací. Typickým příkladem v souvislosti s vektorovými daty je například Mapbox Studio. V případě vizualizace zpracovávané v rámci této práce je však vytvořen vlastní server distribuující zmíněná vektorová data (viz dále v textu), tudíž není potřeba dané limity respektovat. Tento přístup s sebou přináší výhodu především do budoucna, kdy se počítá s aktualizací vizualizace. Vzhledem k neustálému růstu dat SPOI by totiž zmíněná omezení vedla k přerušení transformace souborů z formátu GeoJSON do MBTiles a nově by se musela nastavit základní úroveň vykreslení i míra generalizace.

Tippecanoe tak představuje výkonný a efektivní nástroj pro tvorbu vektorových mapových dlaždic disponující širokou škálou možností nastavení převodu a schopností zpracovávat obrovská množství bodů. Jeho nevýhodou je pouze neexistující podpora pro systémy Windows. Standardní instalace jako balíčku je dostupná pouze pro macOS přes správce Homebrew. Možnost tento nástroj spustit poté existuje i na linuxové distribuci, nicméně uživatel si jej musí nejdříve sám ručně zkompilovat. Další možnou nevýhodou je neexistence grafického prostředí, s nástrojem se tedy pracuje pouze ve formě příkazové řádky.

Dalším stěžejním krokem v procesu tvorby aplikace bylo vytvoření vlastního serveru, který bude vektorové mapové dlaždice distribuovat a umožní tak jejich využití v koncových aplikacích. Pro tento účel lze sice zdarma využít služeb společnosti Mapbox nabízející kartografické nástroje, styly a hostingy speciálně navržené pro prostorová data ve vektorové podobě, nicméně k tomu nebylo přistoupeno. Důvodem jsou drobná technická a právní omomezení vztahující se k výsledné aplikaci a především proprietární řešení. Využit tak byl open-source nástroj TileServer PHP. Ten je snadno spustitelný na téměř jakémkoliv standardním webhostingu založeném na Apache HTTP Serveru, který disponuje PHP včetně SQLite modulu a podporuje uživatelské úpravy modulu "mod_rewrite", sloužícímu k modifikaci URL adres, a souboru ".htaccess", který uživateli umožňuje měnit některé vlastnosti serveru bez toho, aniž by o to žádal správce. Na tento server byly následně pomocí protokolu FTP nahrány soubory MBTiles získané nástrojem Tippecanoe. TileServer tyto soubory zpracovává a v reálném čase vyřizuje požadavky aplikací na přenos konkrétních vektorových mapových dlaždic, které poskytuje po síti ve formátu PBF. Připojení na server přitom z pohledu aplikace probíhá přes URL, která je sestavena obdobně jako u rastrových mapových podkladů OpenStreetMap či jiných obdobných poskytovatelů. Pro každou vrstvu dat SPOI je tak vytvořena konkrétní adresa[33], přes kterou lze data do aplikace integrovat. Výhodou poskytování vektorových mapových dlaždic zmíněným způsobem je možnost využít takto zpracovaná data SPOI i v jiných aplikacích než je pouze vizualizace vytvářená v rámci této práce. Navíc je možné připojit do řešení třetích stran pouze jednu či více konkrétních kategorií a není tedy potřeba vždy pracovat s celou poměrně objemnou datovou sadou.

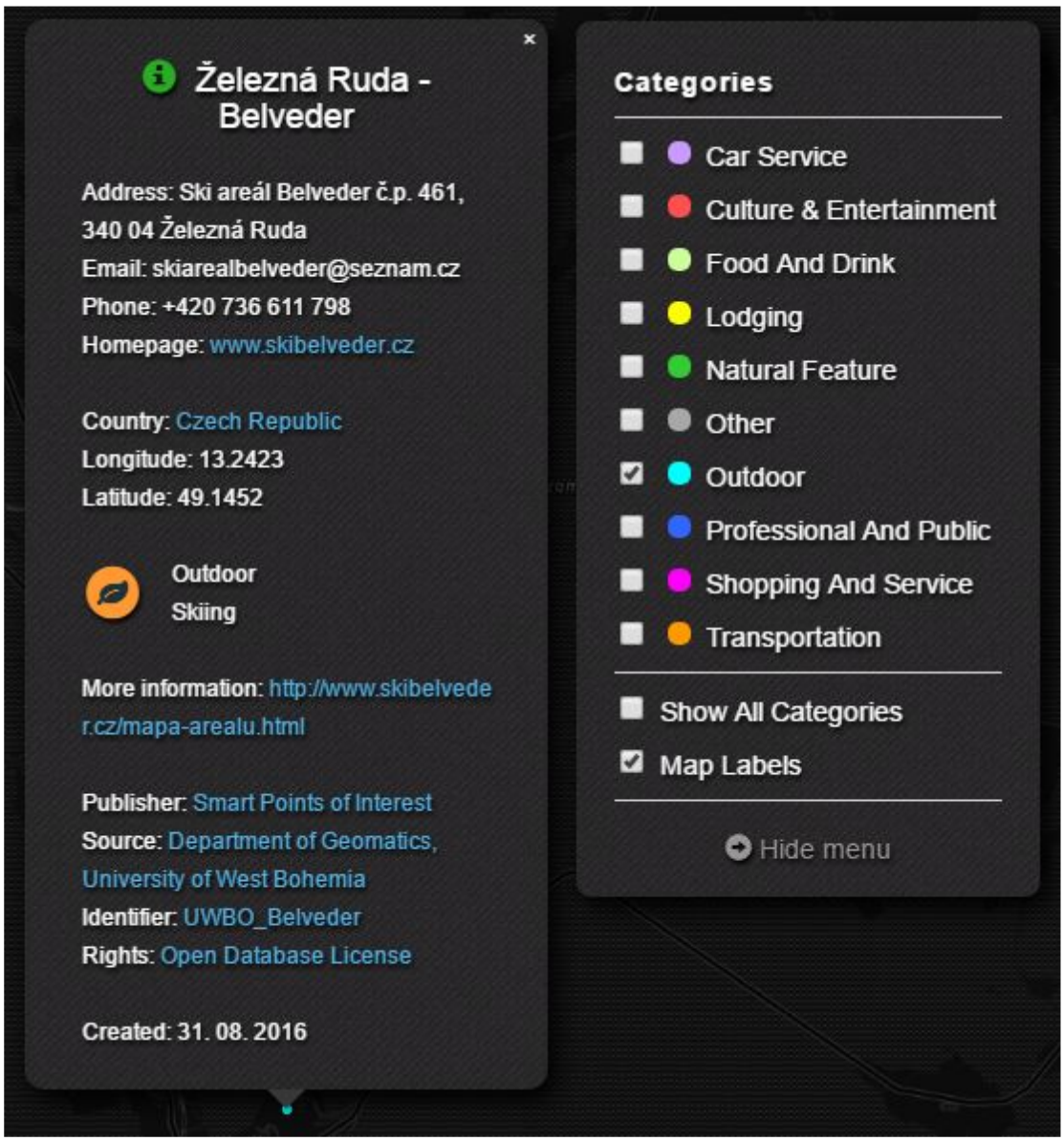
[33] Například kategorie Car Service má neměnnou URL
http://jachymkellar.eu/tileserver/SPOI1/{z}/{x}/{y}.pbf

Další kroky v procesu vizualizace dat SPOI jsou poté spojeny již přímo s tvorbou samotné interaktivní webové aplikace. Ta je vytvářena pomocí javascriptové open-source knihovny Mapbox GL JS, která je zaměřena přímo na tvorbu interaktivních map z vektorových podkladů. Pro jejich vykreslení navíc využívá technologii WebGL, což má pozitivní vliv na rychlost celé aplikace. Kromě nastavení všech vrstev a jejich napojení na zmíněný TileServer je na tomto místě také definován vzhled a parametry vykreslení jednotlivých bodů pomocí Mapbox Style Specification. Jedná se o JSON objekt, který definuje co, jak a v jakém pořadí kreslit. Konkrétně jde jednak o barevné rozlišení jednotlivých vrstev, ale především o nastavení velikosti teček, která byla definována jako funkce v závislosti na úrovni přiblížení. Jejich velikost se tedy zvětšuje s každou změnou úrovně směrem do větších měřítek. Opět se zde však naráží na problém jejich překrývání a snahou mu předejít zmenšením tečky na straně jedné a na straně druhé snahou prezentovat tečku jako co největší, aby na ni šlo jednoduše kliknout. Výsledná velikost, která byla zvolena na základě testování, je poté poměrně malá. Cílem je tedy předejít především nečitelnosti způsobené výrazným překryvem a snahou zachovat geometrickou přesnost. Zároveň však byla ale v kódu zvětšena aktivní oblast kolem každé tečky, takže se informace o daném bodě zobrazí i po kliknutí několik pixelů mimo daný bod. Symbol kurzoru je přitom nastaven pro celou aktivní oblast jako pro klasický odkaz.

Dále je z knihovny Mapbox GL JS využita funkce “queryRenderedFeatures”, která umožňuje přistupovat ke všem atributům, které jsou s jednotlivými body spojeny. Po kliknutí na příslušnou tečku se tedy za běhu (tzv. “on the fly”) vytvoří objekt (DOM), do kterého se nahrají všechny existující popisné atributy, a ten se následně vloží do vyskakovacího okna (též zvaného popup), které je umístěno na souřadnicích zvolené tečky. Jednotlivé informace jsou přitom upravovány tak, aby byly co nejvíce uživatelsky přívětivé. To znamená, že například popisek ve tvaru “caffé_muzeum” se automaticky změní na “Caffé Muzeum”. Seznam všech atributů z datového modelu SPOI, které se při výskytu v datech automaticky zobrazí i v aplikaci je poté součástí příloh. Ukázka vyskakovacího okna se zmíněnými informacemi vygenerovaného přímo ve výsledné vizualizaci je zachycena na obrázku 11.

Do aplikace bylo také následně implementováno menu umožňující třídit jednotlivé body SPOI podle deseti zmíněných kategorií (viz obr. 11). Jeho součástí je i možnost zobrazit či skrýt všechny vrstvy najednou a možnost zobrazit v aplikaci mapové popisky

jako jsou názvy kontinentů či států. Celá vizualizace přitom využívá rastrové podkladové mapy z platformy CARTO založené na datech OpenStreetMap a laděné do tmavě šedé až černé barvy. Tím je docíleno vyniknutí samotných bodům SPOI. Dalším prvkem aplikace je poté možnost přepnout mapu do režimu na celou obrazovku či možnost zobrazení informací o vizualizaci a použitých nástrojích. Součástí je i odkaz na statickou mapu vytvořenou v rámci kapitoly 5.2.




Obr. 11 Ukázka zobrazení popisných atributů při kliknutí na příslušný bod (vlevo) a menu umožňující třídit body SPOI podle deseti základních kategorií (vpravo).

Pro vzhled a grafickou úpravu celé aplikace byl použit open-source front-endový framework Bootstrap a pro vektorové ikony toolkit Font Awesome. Některé efekty a funkce, jako například pozvolné zmizení či zobrazení menu, je řešeno javascriptovou knihovnou jQuery.

## 5.4 Analýza vytvořené interaktivní aplikace

Výsledná interaktivní webová aplikace[34] jako hlavní výstup této diplomové práce tedy prezentuje všechny body SPOI a navíc umožňuje zobrazit i jejich veškeré popisné atributy. Její hlavní předností je především rychlost. Řešení přes vektorové mapové dlaždice, které jsou navíc vykreslovány technologií WebGL, totiž umožňuje načíst celou aplikaci i s body v čase kolem jedné sekundy. Při následném posouvání mapy a jejího přiblížení do větších měřítek jsou časy vykreslení ještě výrazně menší a to i přesto, že aplikace pracuje s téměř 29 miliony bodů[35]. Přesnou hodnotu latence však nelze stanovit, jelikož jsou data SPOI vykreslována až na straně klienta a záleží tedy nejen na parametrech jeho grafické karty, ale například i a rychlosti připojení k internetu. Nicméně si lze všimnout, že se ve větších měřítkách vektorová data SPOI načítají a rovnou i vykreslují i mnohonásobně rychleji než podkladové rastrové mapové dlaždice.

Další nespornou výhodou řešení pomocí převodu dat do vektorů je značná redukce potřeby přenosu velkých objemů dat po síti. Jiné přístupy a konkrétně i ukázkové aplikace vytvořené v kapitole 4.3 totiž posílají ke klientovi i zdrojová data (nejčastěji ve formátu CSV), která jsou zde následně vykreslována. Nicméně i pokud je použita například technologie WebGL a uživatel disponuje rychlým připojením, zabere příliš dlouhý čas zdrojové soubory vůbec zpracovat. Jejich velikost se totiž u většího počtu bodů pohybuje i v řádu několik desítek megabajtů či ještě více. Na druhou stranu řešení použité v této práci využívá již zmíněné vektorové mapové dlaždice, které jsou přes TileServer distribuovány ve formátu PBF. U naprosté většiny z nich se však jejich velikost pohybuje pouze v řádu desítek kilobajtů. Celá aplikace tak při prvotním načtení pracuje jen s velikostí kolem jednoho megabajtu a to jsou do této hodnoty započítány i velikosti

[34] Vizualizace je umístěna na adrese: http://jachymkellar.eu/spoi/. Zdrojové kódy a návod na její spuštění na jiné doméně či vlastním počítači jsou poté součástí přiloženého DVD. Ukázka zobrazení všech bodů všech kategorií pro oblast Evropy je zachycena na obrázku 12.

[35] V datech SPOI použitá klasifikační ontologie umožňuje zařadit jeden bod do více kategorií. Ve vytvořené aplikaci jsou tedy takové body zahrnuty zvlášť ve všech vrstvách (kategoriích), do kterých patří.

podkladových rastrových mapových dlaždic a zdrojové soubory použitých knihoven a stylů.

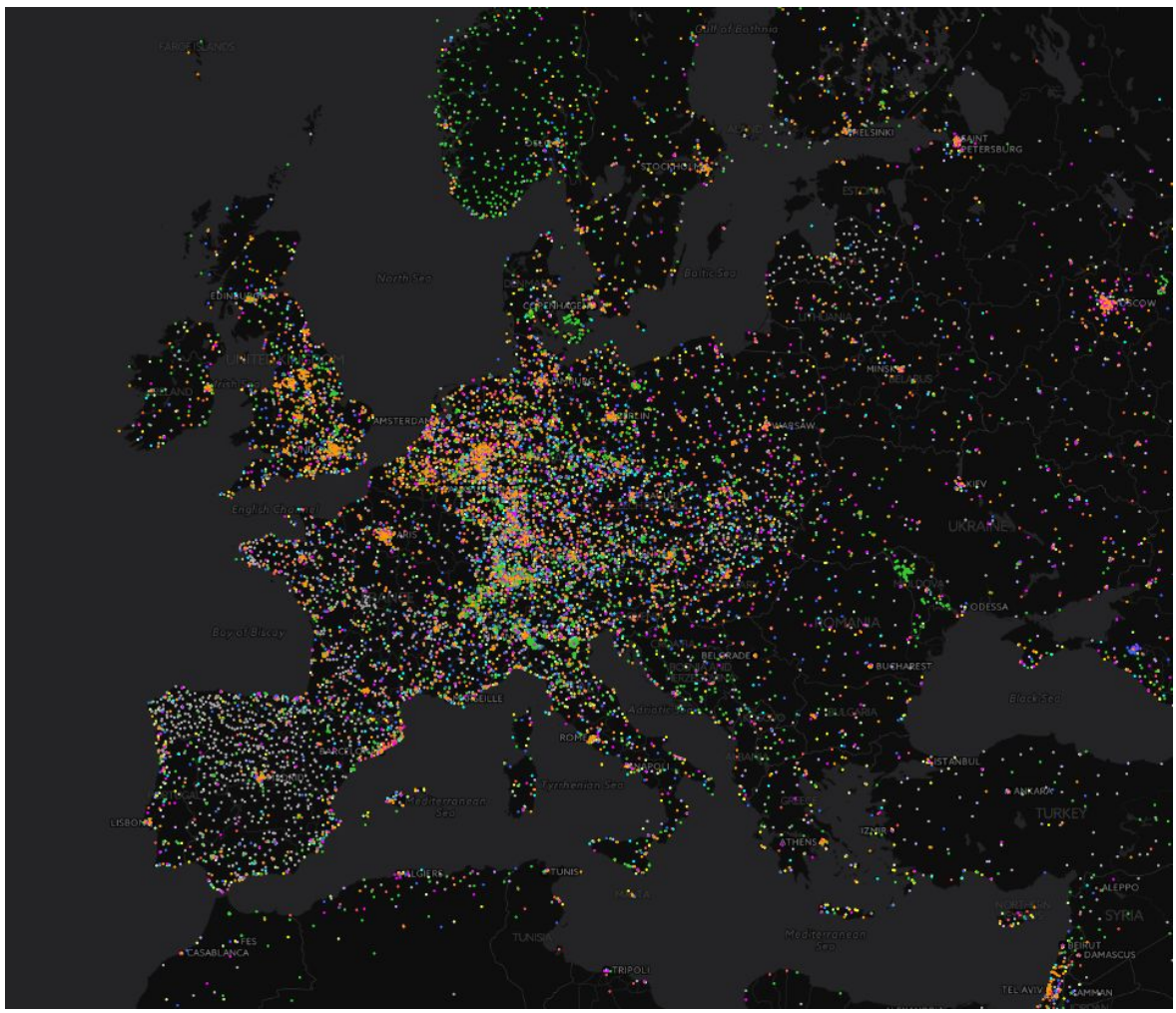
Vytváření objektů (DOM) za běhu podle potřeby aplikace přímo z vektorového podkladu navíc umožňuje přistupovat ke všem popisným atributům všech bodů bez jakékoliv ztráty na výkonu. Jinými slovy je objekt pro požadovaný bod vždy vytvořen až po kliknutí na příslušnou tečku v mapě. Na rozdíl od již existujícího mapového klienta zpracovaného pomocí HSLayers-NG (viz kapitola 5.1) či ukázkových aplikací z kapitoly 4.3 tak nejsou vytvářeny objekty pro každý bod ještě před načtením celé aplikace nebo pro každý bod z aktuálního mapového pohledu. Webový prohlížeč tedy nemusí zpracovávat velké množství objektů a nehrozí tedy jeho zamrznutí či pomalejší načítání dat.

Další výhodou představené vizualizace jako celku od zpracování až po výslednou reprezentaci je využití pouze open-source nástrojů, knihoven a formátů. Není dokonce vyžadován ani přístupový token či vodoznak společnosti Mapbox, jelikož od ní nejsou využity žádné mapové styly, mapové dlaždice ani hosting a žádná část vizualizace není vytvořena pomocí aplikace Mapbox Studio. Použity jsou od této společnosti pouze nástroje a formáty (Tippecanoe, MBTiles, Mapbox GL JS), které jsou distribuovány jako open-source bez jakýchkoliv omezení. Jediným nástrojem bez otevřené specifikace je HJ-Split, který je dostupný jako freeware. Jeho použití zdarma je však možné bez omezení jak pro soukromé tak i firemní účely. Není pouze dovoleno žádat za tento program jako takový o peníze. Ve vizualizaci jsou také využity rastrové podkladové mapy z platformy CARTO od společnosti CartoDB, Inc. Ty lze zdarma využít pro jakékoliv nekomerční a nesoukromé aplikace, u kterých není přesažen limit 75 000 mapových zobrazení za měsíc. V opačném případě by bylo nutné získat klíč a přejít na placenou verzi. Tyto podklady však nejsou pro aplikaci ani samotnou tvorbu vizualizace stěžejní, zvoleny byly pouze z vizuálního důvodu, jelikož dávají díky kombinaci pouze černých a šedých odstínů vyniknout bodům SPOI. V kódu aplikace lze však tento zdroj změnit přepsáním jediného řádku a je tedy možné využít například podkladové rastrové mapy OpenStreetMap či vytvořit vlastní verzi těchto map založenou na zmíněné kombinaci barev. Tento úkol však již není cílem této práce.

Další výhodou vytvořené vizualizace jako celku je responzivní design. S aplikací lze tedy pracovat jak na počítačích, tak i na tabletech či mobilních telefonech.

Jedinou podmínkou je nutnost disponovat webovým prohlížečem, který podporuje WebGL. Nicméně tato technologie je v prohlížečích pro desktopy i telefony již běžně implementována jako standardní výbava. Jako zajímavost lze poté označit možnost nejen mapou posouvat, ale i s ní rotovat (držením pravého tlačítka myši). Tato funkce je standardně zabudována v Mapbox GL JS a slouží především k rotaci pohledu při vykreslování 3D objektů.

Výsledná aplikace byla také analyzována z pohledu použitelnosti a správné funkčnosti. Testována tak byla například validita jejího HTML i CSS kódu[36], responzivní design[37], kompatibilita napříč různými webovými prohlížeči[38] či správná syntaxe kódu javascriptu[39].




Obr. 12 Ukázka zobrazení bodů SPOI všech kategorií pro oblast Evropy.

[36] Markup Validation Service: https://validator.w3.org
CSS Validation Service: https://jigsaw.w3.org/css-validator
[37] Například službou Am I Responsive? http://ami.responsivedesign.is/
[38] Turbo.net Browser Sandbox: https://turbo.net/browsers
[39] Esprima: Syntax Validator: http://esprima.org/demo/validate.html

Na tomto místě je také nutné zdůvodnit lokalizaci výsledné webové aplikace do anglického jazyka. Ten byl vybrán především kvůli skutečnosti, že se nejedná o vizualizaci, která je vytvořena pouze jako ukázka pro účely této práce, ale má být reálně využita pro prezentaci dat SPOI a umožnit tak všem uživatelům tato data prohlížet. Zmíněná datová sada však obsahuje většinu dat právě v angličtině a navíc je vytvářena a spravována v rámci mezinárodních projektů. Snahou je tedy neomezovat se pouze na českou lokalizaci a nabídnout možnost prozkoumat data mnohem většímu okruhu uživatelů a to právě díky použití anglického jazyka.

Za nevýhodu aplikace lze poté označit nutnost její manuální aktualizace. Na rozdíl od již zmíněného mapového klienta řešeného přes HSLayers-NG nepracuje vytvořená vizualizace s databázovým enginem Virtuoso, ale přímo se zdrojovými RDF soubory, čímž je umožněno dosáhnout všech výhod popsaných výše v textu. Při aktualizaci datové sady SPOI, která probíhá 4 x ročně právě přes službu Virtuoso, však díky tomu není možné změny v podobě nových bodů či upravených informací u těch stávajících automaticky promítnout i do vytvořené aplikace. Ta tak nemusí být vždy v souladu s aktuální verzí dat SPOI, která je uživatelům dostupná. Případnou aktualizaci vizualizace je poté nutné řešit manuálně a je při ní nutné získat od poskytovatele zdrojová data SPOI ve formátu RDF, která nejsou běžně dostupná v rámci internetu, a zopakovat celý postup zmíněný v kapitole 5.3. Časová náročnost takového úkonu je přitom odhadována na 12 hodin, přičemž naprostou většinu tohoto času zabere běh skriptu převádějící RDF soubory do formátu GeoJSON. Postup aktualizace včetně příkazů a nastavení jednotlivých nástrojů je součástí přiloženého DVD.

# 6. Závěr

Cílem diplomové práce bylo představit a zhodnotit možnosti vizualizace prostorových Big Data a na základě této rešerše následně vhodným způsobem vizualizovat datovou sadu Smart Points of Interest. Nejprve tak bylo nutné zabývat se vymezením samotného pojmu Big Data. Tedy jak ho lze vůbec definovat a jaká data lze označit jako Big Data. Představeny a rozebrány tak byly vlastnosti charakterizující Big Data včetně dalších i kritických pohledů na daný pojem. Celkově lze přitom říci, že je jeho vymezení a definice značně nejednoznačná a existuje více možností, jak na něj nahlížet. Jeho nejdůležitějším aspektem jsou samotná data, s kterými obecně nelze pracovat běžnými nástroji v rozumném čase. Podrobněji jsou poté Big Data charakterizována především velkým objemem s jeho neustálým růstem, požadavkem na jejich co nejrychlejší zpracování a svoji různorodostí, jelikož se jedná především o nestrukturovaná a semistrukturovaná data (tzv „3V“ model). Existuje a neustále přibývá i více vlastností, nicméně ty se setkávají s ještě častější kritikou, a lze je tedy chápat spíše jako upozornění poukazující na problémy a skutečnosti, které s sebou Big Data mohou přinášet, a nad kterými je potřeba se před jejich zpracováním zamyslet. Jejich příkladem může být věrohodnost, platnost nebo pomíjivost. Důležité však je, že není dáno, kolik těchto vlastní musí data splňovat a především v jaké míře, aby se dala označit jako Big Data. Takové určení je tak velice subjektivní a s trochou nadsázky se dá říci, že si každý může jako Big Data označit jakoukoliv datovou sadu, kterou považuje za objemnější, než pro něj bývá běžné. Pojem Big Data navíc není pouze o datech, ale i o architekturách a technologiích, která s nimi pracují. Zmínit je poté nutné i názory, že se jedná o pouhý marketingový pojem.

Dále jsou v práci zmíněna specifika Big Data obsahující prostorovou složku. Především zdroje takových dat, speciální funkce a algoritmy sloužící pro jejich ukládání, analyzování a zpracování jako jsou prostorové datové typy, indexy či operace, informace, které v sobě obsahují (prostorové, popisné a časové), jejich nejčastější formáty a v neposlední řadě i oblasti využití.

Pro následné zhodnocení možností vizualizace bylo také nutné věnovat pozornost jejím jednotlivým částem. Představeno a popsáno tak bylo sedm základních kroků

v procesu vizualizace. Konkrétně se jedná o získávání dat, jejich analýzu, filtraci, dolování informací, samotnou reprezentaci a následnou fázi jejího vylepšování a případné interakce. Shrnuty byly také výhody vizualizace, přičemž ta nejzásadnější je prezentace velkého množství dat člověku srozumitelným způsobem. Vizualizace zároveň slouží i jako prostředek k poznávání a pochopení dosud neznámých skutečností, hledání nových vzorů a umožňuje uživateli snadno vniknout do problému. S ohledem na prostorovou složku je poté stěžejní získání informací o prostorovém rozložení dat, a tedy i zkoumaného jevu. Dále jsou také vymezeny problémy, které je nutné u vizualizace ve spojitosti s Big Data řešit. První se týká především lidského vnímání a souvisí s příliš velkým grafickým zatížením. Při práci s velkým množství dat totiž snadno může dojít k naprosté nepřehlednosti a nečitelnosti výsledné vizualizace. Druhý problém poté vychází ze samotných vlastností Big Data a týká se především rychlosti vykreslení vizualizace a případného interaktivního průzkumu dat uživatelem. Zpracování a výpočet vizualizace Big Data je totiž natolik složitý, že je nemožné ho dosáhnout v potřebném čase běžnými technologiemi. V případě interaktivní aplikace, která tato data prezentuje, je navíc velice složité dosáhnout minimální latence a umožnit tak uživateli s aplikací plynule pracovat.

Dále je v práci provedena samotná rešerše možností vizualizace prostorových Big Data. Tedy jak k tomuto problému vůbec přistupovat, jak ho řešit a jaké konkrétní nástroje lze využít. Tyto možnosti jsou přitom především z důvodu nejasností ve vymezení pojmu Big Data rozděleny do 3 základních kategorií. Vzhledem k vlastnostem Big Data, popsaných v druhé kapitole této práce, je jasné, že je nelze zpracovávat na jednom počítači. První kategorie se tedy zabývá distribuovanými systémy a s nimi spojenými paralelními výpočty. Ty využívají diskové a výpočetní kapacity více strojů a zpracování vizualizace, které by například na jednom počítači trvalo několik dní nebo by ani nebylo řešitelné, je tak díky těmto systémům možné vyřešit v rámci minut nebo případně i v reálném čase. Tato kategorie tak umožnuje zpracovávat „pravá“ Big Data v řádu až terabajtů, petabajtů a teoreticky i větších objemů dat. Součástí rešerše je poté i popis a princip funkce takových systémů. Představen je tak Apache Hadoop, Apache Spark a další frameworky či databáze z tzv. Hadoop ekosystému. Nad nimi jsou poté jako rozšíření vytvářeny nástroje, které umožňují práci s prostorovými Big Data včetně jejich vizualizace, a které jsou tedy z hlediska prováděné rešerše zcela zásadní. Konkrétně se diplomová práce zabývá nástroji SpatialHadoop, GeoTrellis, ESRI GIS Tools

for Hadoop, GeoMesa, Babylon a GeoWave. Součástí je i základní popis těchto nástrojů včetně jejich porovnání a shrnutí výhod, nevýhod či vhodnosti a náročnosti použití. Z hlediska vizualizace je však jejich nabídka značně omezená. Důvodem je především skutečnost, že se jedná o poměrně nové nástroje, které se teprve rozvíjejí. Představují tedy spíše potenciál do budoucna a to i díky open-source licencím. Celkově lze poté vizualizaci prostorových Big Data za pomoci distribuovaných systémů a paralelních výpočtů hodnotit jako značně složitou a náročnou a to jak z hlediska potřebných znalostí, tak i potřebného technického vybavení.

Druhá kategorie se poté zabývá cloudem a obecně službami třetích stran. Uživatel si tedy pronajímá jejich distribuované systémy, úložiště či výpočetní kapacity. Jedná se o nejsnazší cestu, jak vizualizovat prostorová Big Data, nicméně je nutné podotknout, že jsou přitom použita proprietární a především placená řešení. I přesto se však jedná o jednu z možných cest, jak prostorová Big Data vizualizovat a je tedy představeno i několik konkrétních poskytovatelů, kteří nabízejí nejen uložení a zpracování Big Dat, ale i přímo jejich vizualizaci.

Třetí kategorie je zaměřena na pokročilé vizualizační nástroje, které představují řešení vizualizace většího množství dat na jednom počítači. Nutno však podotknout, že u této kategorie jednak existuje horní hranice velikosti dat, která lze takto zpracovat, a jednak už je trochu sporné, zda se jedná o Big Data, nebo ne. Konkrétně je poté představena technologie WebGL a na ní založené nástroje pro vizualizaci prostorových dat, dále knihovny Supercluster a Prunecluster využívající shlukování prostorově si blízkých bodů, čímž eliminují počet zobrazovaných prvků, nástroj Datashader v kombinaci s webovým prostředím Jupyter Notebook a desktopový software Tableau Public. Zmíněnými nástroji je navíc vytvořena demonstrativní aplikace porovnávající jejich možnosti a výkon. V této kategorii je dále popsána možnost řešení vizualizace prostorových Big Data pomocí nástrojů generující z dat mapové dlaždice a to jak v rastrové, tak i vektorové formě. Oba tyto přístupy jsou zhodnoceny a jsou popsány jejich výhody a nevýhody. Na základě celé rešerše je poté vybráno řešení pomocí vektorových mapových dlaždic jako nejvhodnější pro vizualizaci datové sady SPOI.

Druhým hlavním cílem diplomové práce je poté prakticky vizualizovat již zmíněnou datovou sadu Smart Points of Interest. Provedena tak byla její analýza a následně bylo rozhodnuto vytvořit jak vizualizaci statickou, tak i ve formě interaktivní

webové aplikace. V prvním případě byla datová sada převedena do formátu HDF5 a následně pomocí programování v jazyce Python a nástroje Datashader vykreslena v prostředí aplikace Jupyter Notebook. Výsledná statická vizualizace tak v malém měřítku zobrazuje všech 27,5 milionů bodů SPOI najednou. Díky aplikaci barevné stupnice je navíc patrná i hustotu rozložení těchto bodů v jednotlivých oblastech světa.

Stěžejním bodem poté bylo vytvoření vizualizace ve formě webové aplikace umožňující interaktivně prozkoumávat všechny body SPOI včetně jejich popisných atributů. Pro tento účel byla datová sada převedena do formátu GeoJSON a následně pomocí nástroje Tippecanoe konvertována do vektorových mapových dlaždic. Dále byl nastaven vlastní server, který tyto dlaždice distribuuje v prostředí internetu. Pomocí knihovny Mapbox GL JS byla poté vytvořena klientská aplikace, která zmíněná data vykresluje pomocí technologie WebGL. Výsledná aplikace tak umožňuje plynulý interaktivní průzkum celé datové sady bez známky vyšší latence, řeší generalizaci v jednotlivých úrovních přiblížení, minimalizuje objem dat přenášených po síti, umožňuje třídit body podle jednotlivých kategorií a po kliknutí na konkrétní bod navíc zobrazuje všechny jeho popisné atributy. Využity jsou přitom pouze open-source nástroje a formáty.

Rešeršní část této práce tak představuje možnosti vizualizace větších objemů prostorových dat či přímo Big Data. Nabízí různé konkrétní nástroje, které lze v závislosti na formě požadované vizualizace a na vlastnostech daných dat použít. Praktická část poté demonstruje a popisuje využití některých nástrojů při řešení vizualizace konkrétní datové sady. Jejím hlavním výstupem je poté interaktivní webová aplikace, která umožňuje zkoumat zmíněná data uživatelsky přívětivým způsobem. Práce tedy splňuje požadavky a cíle stanovené v jejím úvodu.

S ohledem na další možné kroky, kterým se lze v souvislosti s vizualizací prostorových Big Data věnovat, jsou autorem navrženy především tři hlavní oblasti. První se týká distribuovaných systémů a možnosti zaměřit se detailněji na jednotlivé vizualizační nástroje. Pro jejich využití v praxi by bylo vhodné je otestovat různě velkými datovými sadami, a především je popsat z uživatelského pohledu. Stručná dokumentace a minimum návodů totiž může mnoho uživatelů odradit od jejich použití a navíc brání i jejich dalšímu rozšíření mezi nové a méně technicky zkušené uživatele. Vzhledem k open-source licencím těchto nástrojů a dostupnosti jejich kódu na platformě GitHub[40] je

[40] https://github.com/

také možné podílet se i přímo na jejich vývoji. Druhá oblast se poté týká vizualizace datové sady SPOI, konkrétně převodu jejích dat z formátu RDF do GeoJSON. Pro tuto konverzi byly vytvořeny skripty v jazyce PHP, které si s ní sice poradí, nicméně jim dané zpracování zabere mnoho času (cca 8 hodin). Ačkoliv je tento převod prováděn offline a neovlivňuje tedy nijak výkon výsledné aplikace, může být předmětem další práce jeho optimalizace, či využití jiných nástrojů nebo programovacího jazyka. S ohledem na grafickou stránku vytvořené vizualizace může být také její podoba vytvořena ve "světlé" variantě. Nabízí se tedy možnost využít například klasické podkladové mapy OpenStreetMap a místo teček zvolit grafické ikony. Třetí oblastí, která může být z pohledu vizualizace prostorových dat a Big Data zajímavá, je práce s nástrojem Tippecanoe. Ten představuje velice efektivní a výkonné řešení převádějící objemné datové sady do vektorových mapových dlaždic. Kromě využité funkce převodu bodů však nabízí i mnoho dalších možností, jako je práce s liniemi a polygony. Je tedy možné pomocí něj vektorově zpracovávat celé kompletní mapové podklady.

# 7. Použitá literatura a informační zdroje

## 7.1 Knižní zdroje a odborné publikace


1. AKERKAR, Rajendra (2013). *Big Data Computing*. CRC Press. 564 s. ISBN 9781466578371.

2. AKHGAR, Babak, SAATHOFF, Gregory, ARABNIA, Hamid, HILL, Richard, STANIFORTH, Andrew, BAYERL, Petra Saskia (2015). *Application of Big Data for National Security: A Practitioner's Guide to Emerging Technologies*. Butterworth-Heinemann. 316 s. ISBN 9780128019733.

3. ANDERSON, Tessa (2009). Kernel density estimation and K-means clustering to profile road accident hotspots. *Accident Analysis & Prevention*. 41 (3). s. 359-364. ISSN 0001-4575

4. ANDREWS, Gregory (2000). *Foundations of Multithreaded, Parallel, and Distributed Programming*. Addison–Wesley. 664 s. ISBN 0-201-35752-6.

5. APARICIO, Manuela, COSTA, Carlos (2014). Data visualization. *Communication Design Quarterly Review.* 3 (1), s. 7 - 11.

6. BRONSON, Jonathan, SUMMA, Brian,COMBA, Joao, FREIRE, Juliana, HOWE, Bill, PASCUCCI, Valerio, SILVA, Claudio (2011). Parallel Visualization on Large Clusters using MapReduce. In: *IEEE Symposium on Large Data Analysis and Visualization*. 23.-24.10.2011, Providence, RI, USA.

7. BUYYA, Rajkumar, CALHEIROS, Rodrigo, DASTJERDI, Amir Vahid (2016). *Big Data: Principles and Paradigms*. Morgan Kaufmann. 494 s. ISBN 9780128093467.

8. COLETTE, Andrew (2013). *Python and HDF5: Unlocking Scientific Data*. O'Reilly Media, Inc. 152 s. ISBN 9781491945018.

9. COULOURIS, George, DOLLIMORE, Jean, KINDLBERG, Tim, BLAIR, Gordon (2011). *Distributed Systems: Concepts and Design*. 5 vyd. Boston: Addison-Wesley. 1047 s. ISBN 0-132-14301-1.

10. ČERBA, Otakar, MILDORF, Tomáš (2017). Smart Points of Interest: Big, Linked and Harmonized Spatial Data. In: *AutoCarto 2016*. 14. - 16.9.2016, Albuquerque, New Mexico, USA.

11. ČERBA, Otakar, CHARVÁT, Karel, MILDORF, Tomáš, BĒRZIŅŠ, Raitis, VLACH, Pavel, MUSILOVÁ, Barbora (2016a). SDI4Apps Points of Interest Knowledge Base. In: *Progress in Cartography*. Springer International Publishing AG. s. 229-237. ISBN 978-3-319-19601-5.

12. ČERBA, Otakar, BĒRZIŅŠ, Raitis, CHARVÁT, Karel, MILDORF, Tomáš (2016b). Smart POI: Open and linked spatial data. In: *European Geosciences Union, General Assembly*. 17. – 22.4.2016, Vídeň, Rakousko.

13. ČERNÝ, Michal (2014). *Big data a jejich možnosti v kontextu knihoven*. Knihovna. 24 (1), s. 104 - 111.

14. DANCHILLA, Brian (2012). *Beginning WebGL for HTML5*. Apress. 356 s. ISBN 9781430239970.

15. DEAN, Jeffrey, GHEMAWAT, Sanjay (2004). MapReduce: Simplified Data Processing on Large Clusters. In: *OSDI'04: Sixth Symposium on Operating System Design and Implementation*. 5.12.2004, San Francisco, CA.

16. ELDAWY, Ahmed, MOKBEL, F. Mohamed (2015). The Era of Big Spatial Data: A Survey. *DBS Journal*. The Database Society of Japan, 13(1), s. 25-36.

17. ELMQVIST, Niklas, FEKETE, Jean-Daniel (2009). Hierarchical Aggregation for Information Visualization: Overview, Techniques, and Design Guidelines. *IEEE Transactions on Visualization and Computer Graphics*. 16 (3). s. 439 - 454.

18. EVANS, Michael, OLIVER, Dev, ZHOU, Xun, SHEKHAR , Shashi (2014). Spatial big data: case studies on volume, velocity, and variety. In: *Big Data: Techniques and Technologies in Geoinformatics*. CRC Press. s. 149 - 176.

19. FRY, Ben (2007). *Visualizing Data*. O'Reilly Media. 384 s. ISBN 978-0-596-51455-6.

20. GREEN, Ido (2012). *Web Workers*. O'Reilly Media, Inc. 46 s. ISBN 9781449322137

21. HASSANIEN, Aboul-Ella, AZAR, Ahmed Taher, SNASEL, Vaclav, KACPRZYK, Janusz, ABAWAJY, Jemel (2015). *Big Data in Complex Systems: Challenges and Opportunities*. Studie in Big Data 9. Springer. 499 s. ISBN 9783319110561.

22. HOLUBOVÁ, Irena, KOSEK, Jiří, MINAŘÍK, Karel, NOVÁK, David (2015). *Big Data a NoSQL databáze*. Grada Publishing a.s. 288 s. ISBN 8024759381.

23. HURWITZ, Judith, NUGENT, Alan, HALPER, Fern, KAUFMAN, Marcia (2013). *Big Data For Dummies*. John Wiley & Sons, Inc. Hoboken, New Jersey. 312 s. ISBN 978-1-118-50422-2.

24. CHEN, Min, MAO, Shiwen, LIU, Yunhao (2014). Big Data: A Survey. *Mobile Networks and Applications*. 19(2), s. 171-209.

25. JEŽEK, Jan, JEDLIČKA, Karel, MILDORF, Tomáš, KELLAR, Jáchym, BERAN, Daniel (2017). Design and Evaluation of WebGL-Based Heat Map Visualization for Big Point Data. In: *The Rise of Big Spatial Data*. Springer International Publishing AG. s. 13-39. ISBN 978-3-319-45122-0.

26. JAIN, Vinay Kumar (2017). *Big Data and Hadoop*. Khanna Publishing. 600 s. ISBN 9789382609131.

27. KELLAR, Jáchym (2015). *Integrace dat pro účely cykloturistické aplikace*. Plzeň. Bakalářská práce. Západočeská univerzita v Plzni. Katedra matematiky. Vedoucí práce Ing. Mgr. Otakar ČERBA, Ph.D.

28. KOCHERLAKOTA, Sarat, HEALEY, Christopher (2005). *Summarization Techniques for Visualization of Large Multidimensional Datasets*. Technical Report TR-2005-35. Knowledge Discovery Lab. Department of Computer Science, North Carolina State University Raleigh, NC 27695-8207.

29. LANEY, Douglas (2001). *3D Data Management: Controlling Data Volume, Velocity and Variety*. Application Delivery Strategies, META Group Inc. File No. 949.

30. LEE, Jae-Gil, KANG, Minseo (2015). Geospatial Big Data: Challenges and Opportunities. *Big Data Research*. Elsevier. 2(2), s. 74-81.

31. LI, Songnian, DRAGICEVIC, Suzana, CASTRO, Francesc Antón, SESTER, Monika, WINTER, Stephan, COLTEKIN, Arzu, PETTIT, Christopher, JIANG, Bin, HASWORTH, James, STEIN, Alfred, CHENG, Tao (2016). Geospatial big data handling theory and methods: A review and research challenges. *ISPRS Journal of Photogrammetry and Remote Sensing*. 115, s. 119–133.

32. LINCH, Nancy (1996). *Distributed Algorithms*. Morgan Kaufmann. 904 s. ISBN 1-55860-348-4.

33. LIU, Zhicheng, JIANG, Biye, HEER Jeffrey (2013). imMens: Real-time Visual Querying of Big Data. In: *Computer Graphics Forum*. Wiley Online Library, 32(3), s. 421-430.

34. MAGOULAS, Roger, LORICA, Ben (2009. Introduction to Big Data. *Release 2.0*. O'Reilly Media, Sebastopol CA. Issue 11. ISSN 1935-9446.

35. MAYER-SCHÖNBERGER, Viktor, CUKIER, Kenneth (2014). *Big Data*. Computer Press, Albatros Media a.s. Překlad: Gorner Jakub. 256 s. ISBN 9788025141908.

36. OLASZ, Angéla, NGUYEN THAI, Binh (2016). Geospatial Big Data processing in an open source distributed computing environment. *PeerJ Preprints* 4:e2226v1.

37. OLSHANNIKOVA, Ekaterina, OMETOV, Aleksand, KOUCHERYAVY, Yevgeni Koucheryavy, OLSSON Thomas (2015). Visualizing Big Data with augmented and virtual reality: challenges and research agenda. *Journal of Big Data*. 2(22). DOI: 10.1186/s40537-015-0031-2.

38. PARISI, Tony (2014). *Programming 3D Applications with HTML5 and WebGL: 3D Animation and Visualization for Web Pages*. O'Reilly Media, Inc. 404 s. ISBN 9781449363956

39. PERROT, Alexandre, BOURQUI, Romain, HANUSSE, Nicolas, LALANNE, Frédéric, AUBER, David (2015). Large Interactive Visualization of Density Functions on Big Data Infrastructure. In: *5th IEEE Symposium on Large Data Analysis and Visualization*, 25.-26. říjen 2015, Illinois, Chicago, United States.

40. PRAVDA, Ján (2006). *Metódy mapového vyjadrovanie: klasifikácia a ukážky*. Bratislava: Geographia Slovaca 21. 127 s. ISSN 1210-3519.

41. SNIJDERS, Chris, MATZAT, Uwe, REIPS, Ulf-Dietrich (2012). "Big Data": Big gaps of knowledge in the field of Internet Science. *International Journal of Internet Science*. 7 (1), s. 1–5.

42. SUTHAHARAN, Shan (2014). Big data classification: problems and challenges in network intrusion prediction with machine learning. *Performance Evaluation Review*. 41(4), s. 70-73.

43. THOMPSON, David, LEVINE, Joshua, BENNETT, Janine, BREMER, Peer-Timo, GYULASSY, Attila, PASCUCCI, Valerio, PÉBAY, Philippe (2011). Analysis of large-scale scalar data using hixels. In: *Proceedings of Symposium on Large Data Analysis and Visualization (LDAV)*. 23.-24. říjen 2011. IEEE. s 23-30. ISBN 9781467301558.

44. ZIKOPOULOS, Paul, EATON, Chris, DEROOS, Dirk, DEUTSCH, Tom, LAPIS, George (2012). *Understanding Big Data*. McGraw Hil. 141 s. ISBN 978-0-07-179053-6.

## 7.2 Elektronické zdroje

35. AGAFONKIN, Vladimir (2016). *Clustering millions of points on a map with Supercluster* [online]. Mapbox [cit. 24.2.2017]. Dostupné z: https://www.mapbox.com/blog/supercluster/

36. BORNE, Kirk (2014). *Top 10 Big Data Challenges – A Serious Look at 10 Big Data V's* [online]. MapR Technologies, Inc. [cit. 9.11.2016]. Dostupné z: http://www.mapr.com/blog/top-10-big-data-challenges-serious-look-10-big-data-vs

36. BORTNIKOV, Edward (2016). *10 Years of Hadoop and its Israeli Pioneering Researchers* [online]. Hadoop at Yahoo [cit. 24.1.2017]. Dostupné z: http://yahoohadoop.tumblr.com/post/153336735536/10-years-of-hadoop-and-its-israeli-pioneering

37. CARTER, Philip (2011). *Big Data Analytics: Future Architectures, Skills and Roadmaps for the CIO* [online]. International Data Corporation [cit. 10.11.2016]. Dostupné z: http://www.sas.com/resources/asset/BigDataAnalytics-FutureArchitectures-Skills-RoadmapsfortheCIO.pdf

38. CLAVERIE-BERGE, Isabelle (2012). *Solutions Big Data IBM* [online]. IBM Corporation [cit. 5.11.2016]. Dostupné z: http://www-05.ibm.com/fr/events/netezzaDM_2012/Solutions_Big_Data.pdf

39. ČERBA, Otakar (2017). Smart Points Of Interest – několik čísel [online]. In: *Otevřená data a otevřený software nejen pro komerční sektor.* 23.1.2017, Praha [cit. 10.3.2017]. Dostupné z: http://www.ccss.cz/wp-content/uploads/2017/01/7_Cerba_1701_Open_data.pdf

40. ČERNÝ, Michal (2013). *Třináct IT trendů v roce 2013 podle IEEE: Internet věcí, big data i soutěž ve spolehlivosti* [online]. Lupa [cit. 28.9.2013]. Dostupné z: http://www.lupa.cz/clanky/trinact-it-trendu-v-roce-2013-podle-ieee-internet-veci-big-data-i-soutez-ve-spolehlivosti/

41. DEMCHENKO, Yuri (2013). Addressing Big Data Issues in the Scientific Data Infrastructure [online]. In: *TERENA Networking Conference (TNC) - Infrastructure issues in Big Data Science*. 3.6.2013, Maastricht [cit. 9.11.2016]. Dostupné z: https://tnc2013.terena.org/includes/tnc2013/documents/bigdata-nren.pdf

42. DEMPSEY, Caitlin (2012). *Where is the Phrase "80% of Data is Geographic" From?* [online]. GIS Lounge [cit. 14.11.2016]. Dostupné z: https://www.gislounge.com/80-percent-data-is-geographic/

43. DEVAN, Ashley (2016). *The 7 V's of Big Data* [online]. Impact Radius [cit. 1.12.2016]. Dostupné z: https://www.impactradius.com/blog/7-vs-big-data/

44. FEW, Stephen (2004). *Tapping the Power of Visual Perception* [online]. Perceptual Edge [cit. 13.1.2017]. Dostupné z: http://www.perceptualedge.com/articles/ie/visual_perception.pdf

45. CHERIAN, Cyril (2013). *Big Data Visualization with D3* [video online]. QBurst [cit. 18.1.2017]. Dostupné z: https://www.youtube.com/watch?v=jIa7Urflz_o

46. JEDLIČKA, Karel, KOLOVSKÝ, František, JEŽEK, Jan, MARTOLOS, Jan, ŠŤASTNÝ, Jan, BERAN, Daniel, HÁJEK, Pavel (2017). Výpočet dopravních intenzit nad Open Transport Map [online]. In: *Otevřená data a otevřený software nejen pro komerční sektor*. 23.1.2017, Praha [cit. 11.4.2017]. Dostupné z: http://www.wirelessinfo.cz/wp-content/uploads/2017/01/9_Jedlicka_et_al_OpenTransportMap_EU.pdf

47. KANDEL, Sean, HEER, Jeffrey (2013). *Visualizing "Big" Data* [online]. Trifacta Inc. [cit 13.1.2017]. Dostupné z: http://www.slideshare.net/SeanKandel/20131024-big-datavisualization/58

48. L'ASTORINA, Edoardo (2015). *Big Data Visualization: Review of the 20 Best Tools* [online]. Dendelion Blu Ltd. [cit. 18.1.2017]. Dostupné z: http://inspire.blufra.me/big-data-visualization-review-of-the-20-best-tools/

49. LURIE, Andy (2013). *39 Data Visualization Tools for Big Data* [online]. ProfitBricks. The IaaS Company [cit. 18.1.2017]. Dostupné z: https://blog.profitbricks.com/39-data-visualization-tools-for-big-data/

50. MORGAN, Timothy Prickett (2016). *MapD GPU Database Looks Forward To Heftier Iron* [online]. The Next Platform [cit. 14.2.2017]. Dostupné z: https://www.nextplatform.com/2016/03/30/mapd-gpu-database-looks-forward-heftier-iron/

51. NOYES, Katherine (2015). *Five things you need to know about Hadoop v. Apache Spark* [online]. InfoWorld. IDG Communications, Inc. [cit. 19.1.2017]. Dostupné z: http://www.infoworld.com/article/3014440/big-data/five-things-you-need-to-know-about-hadoop-v-apache-spark.html

52. RIJMENAM, Mark van (2016). *Why The 3V's Are Not Sufficient To Describe Big Data* [online]. Datafloq [cit. 1.12.2016]. Dostupné z: https://datafloq.com/read/3vs-sufficient-describe-big-data/166

53. SAS Institute (2013). *Data Visualization: Making Big Data Approachable and Valuable* [online]. Market Pulse: White Paper [cit. 2.12.2016]. Dostupné z: http://www.sas.com/resources/asset/SASCIOMarketPulseDataVizWhitePaper.pdf

54. SHAH, Dhaval (2014). *Millions of data points flying in tight formation* [online]. Aerospace Manufacturing and Design [cit. 28.9.2016]. Dostupné z: http://www.aerospacemanufacturinganddesign.com/article/millions-of-data-points-flying-part2-121914/

# Přílohy

## Seznam příloh

Příloha 1: Seznam softwaru, nástrojů, programových balíčků a knihoven zmíněných v rámci diplomové práce

Příloha 2: Demonstrativní aplikace pro srovnání vizualizačních nástrojů z kapitoly 4.3

Příloha 3: Přehled vlastností dat SPOI zobrazujících se v interaktivní webové aplikaci vytvořené v rámci kapitoly 5.3

Příloha 4: Obsah přiloženého DVD

Příloha 5: Spuštění aplikací ze zdrojových kódů na přiloženém DVD

## Příloha 1: Seznam softwaru, nástrojů, programových balíčků a knihoven zmíněných v rámci diplomové práce

| Název | URL | Verze |
|---|---|---|
| Apache Hadoop | http://hadoop.apache.org/ | 2.7.3 |
| Apache Hbase | https://hbase.apache.org/ | 2.0.0 |
| Apache Hive | https://hive.apache.org/ | 2.1.1 |
| Apache HTTP Server | https://httpd.apache.org/ | 2.4.25 |
| Apache Sentry | https://sentry.apache.org/ | 1.7.0 |
| Apache Accumulo | https://accumulo.apache.org/ | 1.8.0 |
| Apache Cassandra | http://cassandra.apache.org/ | 3.9 |
| Apache Kafka | https://kafka.apache.org/ | 0.10.1.1 |
| Apache Pig | https://pig.apache.org/ | 0.16.0 |
| Apache Spark | http://spark.apache.org/ | 2.1.0 |
| ArcGIS for Desktop: ArcMap | http://desktop.arcgis.com/en/arcmap/ | 10.4.1 |
| ArcGIS Online | http://www.arcgis.com/features/index.html | - |
| ArcGIS Server | http://server.arcgis.com/en/ | - |
| Blaze | http://blaze.pydata.org/ | 0.11.0 |
| Bootstrap | http://getbootstrap.com/ | 3.3.7 |
| Cesium | https://cesiumjs.org/ | 1.30 |
| colorcet | https://github.com/bokeh/colorcet | 0.9.1 |
| Datamaps | https://github.com/ericfischer/datamaps | - |
| Datashader | https://github.com/bokeh/datashader | 0.4.0 |
| D3 (Data-Driven Documents) | https://d3js.org/ | 4.6.0 |
| Font Awesome | http://fontawesome.io/ | 4.7.0 |
| GeoJSON-VT | https://github.com/mapbox/geojson-vt | 2.4.0 |
| GeoMesa | http://www.geomesa.org/ | 1.3.0 |
| GeoServer | http://geoserver.org/ | 2.10.1 |
| GeoSpark | http://geospark.datasyslab.org/ | 0.5.0 |

| | | |
|---|---|---|
| GeoTools | http://www.geotools.org/ | 12.5 |
| GeoTrellis | http://geotrellis.io/ | 1.0.0 |
| GeoWave | https://ngageoint.github.io/geowave/ | 0.9.3 |
| GIS Tools for Hadoop | https://esri.github.io/gis-tools-for-hadoop/ | 2.0 |
| h5py | http://www.h5py.org/ | 2.6.0 |
| HJ-Split | http://www.hjsplit.org/ | 3.0 |
| HPCC | https://hpccsystems.com/ | 6.2.10 |
| Homebrew | https://brew.sh/ | - |
| HSLayers-NG | https://ng.hslayers.org/ | - |
| jQuery | https://jquery.com/ | 3.2.0 |
| Jupyter Notebook | http://jupyter.org/ | 4.2.0 |
| Leaflet | http://leafletjs.com/ | 1.0.3 |
| Leaflet.heat | https://github.com/Leaflet/Leaflet.heat | 0.2.0 |
| LocationSpark | https://github.com/merlintang/SpatialSpark | - |
| Mapbox GL JS | https://github.com/mapbox/mapbox-gl-js | 0.32.1 |
| Mapbox Studio | https://www.mapbox.com/mapbox-studio/ | - |
| Maperitive | http://maperitive.net/ | 2.4.1 |
| Mapnik | http://mapnik.org/ | 3.0.12 |
| Odo | https://github.com/blaze/odo | 0.5.0 |
| OpenLayers | https://openlayers.org/ | 4.0.1 |
| OSM Buildings | http://osmbuildings.org/ | 3.1.0 |
| pandas | http://pandas.pydata.org/ | 0.19.2 |
| Postgres-XL | http://www.postgres-xl.org/ | 9.5 |
| PruneCluster | https://github.com/SINTEF-9012/PruneCluster | 1.0.0 |
| Redash | https://redash.io/ | - |
| SpatialHadoop | http://spatialhadoop.cs.umn.edu/ | 2.4.2 |
| SpatialSpark | http://simin.me/projects/spatialspark/ | 1.0 |
| STARK | https://github.com/dbis-ilm/stark | - |

| | | |
|---|---|---|
| Supercluster | https://github.com/mapbox/supercluster | 2.3.0 |
| Tableau Public | https://public.tableau.com | 10.1 |
| TileMill | https://tilemill-project.github.io/tilemill/ | 0.10.1 |
| TileServer PHP | https://github.com/klokantech/tileserver-php | 2.0 |
| Tippecanoe | https://github.com/mapbox/tippecanoe | 1.16.11 |
| OpenLink Virtuoso | https://virtuoso.openlinksw.com/ | 7.2 |
| Vim editor | http://www.vim.org/ | 8.0.69 |
| Vizicities | http://vizicities.com | 0.3 |
| WebGLayer | http://webglayer.org/ | 2.0 |
| WebGL Earth | http://www.webglearth.com | 2.4.1 |

## Příloha 2: Demonstrativní aplikace pro srovnání vizualizačních nástrojů z kapitoly 4.3

WebGL: http://home.zcu.cz/~kellar/DP/webGL/

PruneCluster: http://home.zcu.cz/~kellar/DP/pruneCluster/

Datashader[41]: http://home.zcu.cz/~kellar/DP/datashader/

Tableau Public: http://home.zcu.cz/~kellar/DP/tableau/

Zdrojové kódy jednotlivých aplikací jsou součástí přiloženého DVD.

Časy v tabulce 2 jsou průměrem dvaceti měření spuštěných na serveru s parametry níže, vždy s vymazáním dat v mezipaměti. Nepředstavují však pouze dobou potřebnou pro samotné vykreslení dat, ale čas načtení aplikace jako celku, tj. zahrnují například i přenos podkladových mapových dlaždic po síti.

Notebook Lenovo IdeaPad Y510p

Operační systém: *Windows 8.1 64-bit*

Procesor: *Intel Core i7 4700MQ @ 2.40GHz*

RAM: *16,0GB Dual-Channel DDR3 @ 817MHz*

Grafika: *2x2047MB NVIDIA GeForce GT 750M*

Monitor: *15,6" Full HD*, *1920x1080@60Hz*

Webový prohlížeč: *Google Chrome v. 55.0.2883.87 m (64-bit)*

[41] Výstupem nástroje Datashader je statický obraz, zmíněná URL adresa tedy obsahuje pouze HTML kopii kódu a výstupu.

## Příloha 3: Přehled vlastností dat SPOI zobrazujících se v interaktivní webové aplikaci vytvořené v rámci kapitoly 5.3

| **Vlastnost** (včetně jmenného prostoru) | **zobrazena v aplikaci** | **Vlastnost** (včetně jmenného prostoru) | **zobrazena v aplikaci** |
|---|---|---|---|
| rdfs:label | *ano* | foaf:mbox | *ano* |
| rdfs:comment | *ano* | poi:fax | *ano* |
| geos:asWKT | *ano* | foaf:homepage | *ano* |
| poi:secondaryGeometry | *ne* | foaf:phone | *ano* |
| poi:region | *ne* | poi:openingHours | *ano* |
| poi:municipality | *ne* | poi:access | *ano* |
| poi:class | *ano* | poi:accessibility | *ano* |
| locn:fullAddress | *ano* | poi:internetAccess | *ano* |
| locn:poBox | *ne* | rdfs:seeAlso | *ano* |
| locn:thoroughfare | *ne* | skos:exactMatch | *ano** |
| locn:locatorDesignator | *ne* | owl:sameAs | *ano** |
| locn:locatorName | *ne* | geos:sfWithin | *ano* |
| locn:addressArea | *ne* | dc:identifier | *ano* |
| locn:postName | *ne* | dc:publisher | *ano* |
| locn:adminUnitL2 | *ne* | dc:title | *ano* |
| locn:adminUnitL1 | *ne* | dc:rights | *ano* |
| locn:postCode | *ne* | dc:source | *ano* |
| locn:addressId | *ne* | dcterms:created | *ano* |

Zde uvedený datový model SPOI a s ním kompatibilní zdrojová data použita pro vizualizaci v rámci této práce jsou aktuální k lednu 2017. Jeho nejnovější verze včetně popisu jednotlivých vlastností a jmenných prostorů je dostupná na adrese http://sdi4apps.eu/spoi/doc/SPOI_data_model.pdf.

Adresa je v aplikaci vyjádřena pouze vlastností locn:fullAddress představující její kompletní popis a ne pomocí jejích jednotlivých komponent (locn:postName, locn:postCode apod.). Informace o státu, ve kterém daný bod leží, je poté extrahována z vlastnosti geos:sfWithin a je k ni automaticky připojen odkaz na stránku anglické verze projektu Wikipedia popisující daný stát (tento odkaz není součástí datového modelu SPOI, ale je generován přímo aplikací).

* U vlastností skos:exactMatch a owl:sameAs je využit pouze odkaz na projekty GeoNames a DBpedia. Odkaz na LinkedGeoData není z uživatelského pohledu zajímavý, a proto není zobrazen, ačkoliv je rovněž součástí vektorových mapových dlaždic.

## Příloha 4: Obsah přiloženého DVD

**Základní adresářová struktura:**

| | |
|---|---|
| .\SPOI - vizualizace interaktivní\ | interaktivní webová vizualizace SPOI (kapitola 5.3) |
| .\SPOI - vizualizace statická\ | statická vizualizace SPOI (kapitola 5.2) |
| .\Ukázkové aplikace\ | aplikace pro srovnání nástrojů z kapitoly 4.3 |
| aplikace.txt | návody pro spuštění aplikací ze zdrojových kódů |
| kellar_dp.pdf | text diplomové práce |
| obsah.txt | obsah DVD |

**.\SPOI - vizualizace interaktivní**

| | |
|---|---|
| .\Aplikace\ | klientská aplikace prezentující body SPOI |
| .\PHP skripty\ | skripty převádějící .RDF soubory do .GeoJSON |
| .\TileServer PHP\ | nastavení serveru distribujícího vektorové mapové dlaždice |
| .\Tippecanoe\ | příkazy pro tvorbu vektorových mapových dlaždic (GeoJSON → MBTiles) |

**.\SPOI - vizualizace statická**

| | |
|---|---|
| .\SPOI.png\ | výsledná statická vizualizace |
| .\Zdrojové kódy\ | zdrojové kódy k vytvoření vizualizace |

**.\Ukázkové aplikace**

| | |
|---|---|
| .\datashader\ | vizualizace pomocí nástroje Datashader |
| .\pruneCluster\ | vizualizace pomocí nástroje PruneCluster |

| | |
|---|---|
| .\tableau\ | vizualizace pomocí nástroje Tableau Public |
| .\webGL\ | vizualizace pomocí WebGL |
| d3.v3.min.js | knihovna D3.js použita pro načítání zdrojových dat |
| data.csv | zdrojová data (extrahována z datové sady SPOI) |

## Příloha 5: Spuštění aplikací ze zdrojových kódů na přiloženém DVD

### Interaktivní vizualizace SPOI

Pro spuštění aplikace je nutné přenést všechny soubory ze složky .\SPOI - vizualizace interaktivní\Aplikace\ na server a následně webovým prohlížečem otevřít soubor index.html. Aplikace vyžaduje přístup k internetu. Webový prohlížeč musí podporovat technologii WebGL.

Pro nastavení vlastního serveru distribujícího vektorové mapové dlaždice (TileServer PHP) stačí přenést všechny soubory z této složky na webhosting založený na Apache HTTP Serveru, který disponuje PHP včetně SQLite modulu a podporuje uživatelské úpravy modulu "mod_rewrite"a ".htaccess". V klientské aplikaci je následně nutné upravit URL jednotlivých vrstev bodů SPOI podle adresy nového serveru.

Již funkční a nastavená aplikace je pro účely vyzkoušení a testování bez potřeby spuštění ze zdrojových kódů umístěna na adrese http://jachymkellar.eu/spoi/. V tomto případě využívá TileServer nastavený na adrese: http://jachymkellar.eu/tileserver/.

### Ukázkové aplikace

Všechny aplikace se spouští otevřením souboru index.html ve webovém prohlížeči. Zdrojové soubory aplikací PruneCluster a WebGL je nutné nejprve přenést na server a poté až spustit. Všechny aplikace vyžadují přístup k internetu. U WebGL musí být použit webový prohlížeč, který tuto technologii podporuje. Aplikace Tableau Public se jako jediná spouští ze serveru třetích stran (viz kapitola 4.3) a není tedy zaručen její bezproblémový chod.