Univerzita Palackého v Olomouci
Přírodovědecká fakulta
Katedra geoinformatiky

VIZUALIZACE GEOGRAFICKY ORIENTOVANÝCH BIG DATA

Bakalářská práce

Ondřej TOMEČKA

Vedoucí práce Mgr. Rostislav Nétek, Ph.D.

Olomouc 2018
Geoinformatika a geografie

ANOTACE

Tato bakalářská práce se v teoretické části zaměřuje na popis charakteristik paradigmatu Big Data. Dále jsou popsány dostupné velkoobjemové datové sady, způsob ukládání velkých objemů dat v distribuovaném souborovém systému, dostupné technologie pro jejich efektivní zpracování, populární distribuce Apache Hadoop frameworku a Hadoop ekosystém jako takový. Pro demonstraci odlišnosti zpracování Big Data oproti konvenčnímu zpracování geodat pracuje autor v praktické části práce s Hadoop distribucí Hortonworks Data Platform a jsou představeny nástroje GIS Tools for Hadoop na příkladu agregace taxi záznamů z New York City do čtvercové sítě. Na tomto příkladu byla otestována i doba potřebná pro dokončení agregace při použití různých rozlišení výsledné čtvercové sítě na dvou výkonnostně odlišných virtuálních počítačích. Další část práce se zaměřuje na vizualizaci dat pomocí JavaScriptových knihoven pro shlukování bodů a tvorbu heatmap a testování, za jak dlouho jsou tyto knihovny schopny vykreslit objemnější datové sady.

KLÍČOVÁ SLOVA

big data; apache hadoop; gis tools for hadoop; vizualizace dat; javascript

Počet stran práce: 49

Počet příloh: 5 (z toho 4 volné a 1 vázaná)

ANOTATION

The theoretical part of this bachelor thesis describes the common characteristics of the Big Data paradigm. It sums up the available data sources of large geospatial datasets, describes how file storage works in distributed file systems, available technologies to effectively process large datasets, the most popular Apache Hadoop distributions and the Hadoop ecosystem as a whole. To demonstrate the differences in processing Big Data compared to conventional processing of geospatial data, the author uses Apache Hadoop distribution Hortonworks Data Platform and GIS Tools for Hadoop to aggregate taxi trip records from New York City into a square grid. The author also tests the impact of using different spatial resolutions of the square bins on the computation time on two virtual machine configurations with different specifications. The next part focuses on data visualization using JavaScript libraries for marker clustering and heatmaps and testing their rendering time when used with larger datasets.

KEYWORDS

big data; apache hadoop; gis tools for hadoop; data visualization; javascript

Number of pages: 49
Number of appendixes: 5

Prohlašuji, že

- bakalářskou práci včetně příloh, jsem vypracoval samostatně a uvedl jsem všechny použité podklady a literaturu.

- jsem si vědom, že na moji bakalářskou práci se plně vztahuje zákon č.121/2000 Sb. - autorský zákon, zejména § 35 – využití díla v rámci občanských a náboženských obřadů, v rámci školních představení a využití díla školního a § 60 – školní dílo,

- beru na vědomí, že Univerzita Palackého v Olomouci (dále UP Olomouc) má právo nevýdělečně, ke své vnitřní potřebě, bakalářskou práci užívat (§ 35 odst. 3),

- souhlasím, aby jeden výtisk bakalářské práce byl uložen v Knihovně UP k prezenčnímu nahlédnutí,

- souhlasím, že údaje o mé bakalářské práci budou zveřejněny ve Studijním informačním systému UP,

- v případě zájmu UP Olomouc uzavřu licenční smlouvu s oprávněním užít výsledky a výstupy mé bakalářské práce v rozsahu § 12 odst. 4 autorského zákona,

- použít výsledky a výstupy mé bakalářské práce nebo poskytnout licenci k jejímu využití mohu jen se souhlasem UP Olomouc, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly UP Olomouc na vytvoření díla vynaloženy (až do jejich skutečné výše).

V Olomouci dne Ondřej Tomečka

Děkuji vedoucímu práce Mgr. Rostislavu Nétkovi, Ph.D. za podněty a připomínky při vypracování práce.

UNIVERZITA PALACKÉHO V OLOMOUCI
Přírodovědecká fakulta
Akademický rok: **2015/2016**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Ondřej TOMEČKA**
Osobní číslo:	**R14510**
Studijní program:	**B1301 Geografie**
Studijní obor:	**Geoinformatika a geografie**
Název tématu:	**Vizualizace geograficky orientovaných Big data**
Zadávající katedra:	**Katedra geoinformatiky**

Zásady pro vypracování:

Cílem práce je provést analýzu paradigmata Big data v oblasti GIS/GIT a příbuzných oborech s ohledem na možnosti vizualizace. Student provede v i mimo oblast GIS/GIT vlastní analýzu, shromáždí a definuje vhodné nástroje, služby, operace, datové sady a jejich poskytovatele, obecné principy, legislativní/licenční omezení. Student vyhledá a analyzuje vhodné zdroje Big data s prostorovým aspektem (NOAA, Landsat, NASA, US Census, ...). Dále teoreticky specifikuje a na reálných případech ověří možnosti vizualizace vybraných zdrojů - konkrétní metody, technologie a postupy prostorových dat. Dále se student v práci zaměří a navrhne ukázkové příklady impelementace nosql databází, nástroje open refine a dalších. Na zvolených případových studiích bude demonstrovat odlišnosti paradigma Big data oproti konvenčnímu zpracování geodat.
Student vyplní údaje o všech datových sadách, které vytvořil nebo získal v rámci práce, do Metainformačního systému katedry geoinformatiky a současně vytvoří zálohu údajů ve formě validovaného XML souboru. Celá práce (text, přílohy, výstupy, zdrojová a vytvořená data, XML soubor) se odevzdá v digitální podobě na CD (DVD) a text práce s vybranými přílohami bude odevzdán ve dvou svázaných výtiscích na sekretariát katedry. O diplomové práci student vytvoří webovou stránku v souladu s pravidly dostupnými na stránkách katedry. Práce bude zpracována podle zásad dle Voženílek (2002) a závazné šablony pro diplomové práce na KGI. Povinnou přílohou práce bude poster formátu A2.

Rozsah grafických prací: **dle potřeby**

Rozsah pracovní zprávy: **max. 50 stran**

Forma zpracování bakalářské práce: **tištěná**

Seznam odborné literatury:

Bedard Y., 2014, Beyond GIS: Spatial On-Line Analytical Processing and Big Data, Univ. of Maine
Chen Y, Suel T., Markowetz A., 2006, Efficient Query Processing in Geographic Web Search Engines
Goodchild M.F. (2013) The quality of big (geo)data. Dialogs in Human Geography 3: 280-284.
Kiehn, W. From big data to data mining (2016) gis.Science - Die Zeitschrift fur Geoinformatik, 3, pp. 18-20.
Pries K.H., Dunnigan R. (2015): Big Data Analytics: A Practical Guide for Managers, 576 p.
Northup P. (2013). Using Big Data in Geographic Information Systems for Observing Earth's Climate. James Madison University, New York, U.S.
Walker J. Big Data Spatial Analytics An Introduction, 35 s., 2013.
Voženílek, Vít. Diplomové práce z geoinformatiky. Univerzita Palackého, 2002.

Vedoucí bakalářské práce: **Mgr. Rostislav Nétek, Ph.D.**
Katedra geoinformatiky

Datum zadání bakalářské práce: **15. června 2016**

Termín odevzdání bakalářské práce: **5. května 2017**

UNIVERZITA PALACKÉHO V OLOMOUCI
PŘÍRODOVĚDECKÁ FAKULTA
KATEDRA GEOINFORMATIKY
17. listopadu 50, 771 46 Olomouc
-1-

L.S.

prof. RNDr. Ivo Frébort, CSc., Ph.D.
děkan

prof. RNDr. Vít Voženílek, CSc.
vedoucí katedry

V Olomouci dne 15. června 2016

OBSAH

SEZNAM POUŽITÝCH ZKRATEK

Zkratka	Význam
AOPK ČR	Agentura ochrany přírody a krajiny České republiky
CDH	Cloudera's Distribution Including Apache Hadoop
CLI	Command Line Interface
CSV	Comma-separated Values
CTO	Chief Technology Officer
ESG	Enterprise Strategy Group
FHV	For-Hire Vehicle
GML	Geography Markup Languague
HDFS	Hadoop Distributed File System
HDP	Hortonworks Data Platform
HiveQL	Hive Query Languague
IOPS	Input/output operations per second
JFK	John F. Kennedy International Airport
JSON	JavaScript Object Notation
KML	Keyhole Markup Languague
LiDAR	Light Detection And Ranging
NASA	National Aeronautics and Space Administration
ND OP	Nálezová databáze ochrany přírody
NFS	Network File System
OGC	Open Geospatial Consortium
PBF	Protocolbuffer Binary Format
POSIX	Portable Operating System Interface
S-JTSK	Systém jednotné trigonometrické sítě katastrální
SHP	Shapefile
SQL	Structure Query Languague
SSH	Secure Shell
UDAF	User-defined Aggregation Function
UDF	User-defined Function
USGS	United States Geological Survey
VGI	Volunteered Geographic Information
VŠB-TUO	Vysoká škola báňská – Technická univerzita Ostrava
WGS84	World Geodetic System 1984
XML	eXtensible Markup Language
YARN	Yet Another Resource Negotiator

ÚVOD

Big Data, pojem, který se během posledních let stal velmi častým předmětem celé řady odborných publikací či vědeckých konferencí. Do dnešní doby stále není určena přesná definice tohoto pojmu a způsoby interpretace se v různých oborech značně odlišují. Obecně lze za Big Data považovat objemné strukturované či nestrukturované datasety, které nemohou být snadno ukládány, spravovány a analyzovány pomocí konvenčních metod v rozumném čase (Chen a kol. 2014).

Obrovský rozmach Big Data je z velké části způsoben klesajícími cenami výpočetní techniky a rozvojem informačních technologi. Dnes již existuje pro zpracování takovýchto dat řada technologií, které jsou neustále vylepšovány. Většina těchto technologií představuje levná řešení pro zpracování velkých objemů dat a mnoho z nich má otevřený zdrojový kód. Příkladem může být framework Apache Hadoop, patřící mezi nejpoužívanější technologie v oblasti Big Data, kombinující běžně dostupný hardware s open source softwarem. Jeho vývoj je podpořen celou řadou přispěvatelů a velkých společností, mezi které můžeme zařadit například Google, Amazon, Microsoft, Facebook či Twitter (Balusamy a kol. 2018).

1 CÍLE PRÁCE

Cílem práce je provést analýzu paradigmata Big data v oblasti GIS/GIT a příbuzných oborech s ohledem na možnosti vizualizace. Student provede v i mimo oblast GIS/GIT vlastní analýzu, shromáždí a definuje vhodné nástroje, služby, operace, datové sady a jejich poskytovatele, obecné principy, legislativní/licenční omezení. Student vyhledá a analyzuje vhodné zdroje Big data s prostorovým aspektem (NOAA, Landsat, NASA, US Census, ...). Dále teoreticky specifikuje a na reálných případech ověří možnosti vizualizace vybraných zdrojů - konkrétní metody, technologie a postupy prostorových dat. Dále se student v práci zaměří a navrhne ukázkové příklady impelementace nosql databází, nástroje open refine a dalších. Na zvolených případových studiích bude demonstrovat odlišnosti paradigma Big data oproti konvenčnímu zpracování geodat.

2 METODY A POSTUPY ZPRACOVÁNÍ

Začátek tvorby bakalářské práce tvořilo podrobné studium literatury týkající se technologií a postupů v oblasti Big Data. Nutné bylo také vyhledat dostupné datové zdroje a zvolit vhodné nástroje a metody pro jejich následnou vizualizaci.

2.1 Použité metody

Marker Clustering

Představuje vizualizační techniku, díky které jsou jednotlivé body na webové mapě seskupeny do shluků podle použitého algoritmu. Takto seskupené body jsou poté reprezentovány na mapě symbolem s počtem bodů, které tvoří shluk. Vytvořeným shlukům lze upravit symbologii podle počtu bodů, které obsahují (např. barvu či velikost). Při přibližování se pak shluky zmenšují a zobrazují se jednotlivé body. Shlukováním se tak eliminují překrývající se body a webová mapa se stává přehlednější.

Heatmap

Jedná se o jednu z nejčastěji používaných metod pro vizualizaci bodových datasetů. Díky této metodě lze snadno spojitě vizualizovat a analyzovat objemné datové sady a identifikovat shluky (Trame a Keßler, 2011). Nelze ale určit, zda jsou tyto shluky statisticky významné či nikoli (Brovelli a kol. 2016). Body jsou reprezentovány barevným přechodem, který představuje oblast a sílu vlivu každého bodu. V případě překryvu pak dochází ke sčítání vlivů těchto bodů (Otte, 2015). Český ekvivalent heatmap stanovila Slézáková (2017) na „mapu či metodu intenzity jevu“, nicméně pro potřeby této práce je použit termín heatmap.

Spatial Binning

Jednu z metod, jak pracovat s objemnými bodovými datasety představuje metoda Spatial Binning. Jedná se o metodu agregování dat do polygonů či do pravidelné sítě, která může být tvořená trojúhelníky, čtverci či v poslední době stále populárnějšími šestiúhelníky. Vytvořené buňky jsou stejně velké tudíž vzájemně srovnatelné a nezávislé na časové a územně proměnlivé administrativní struktuře. Takto vytvořená pravidelná síť umožňuje také hierarchizaci prostorové prezentace a její přesnosti tím, že lze volit velikost buňky při zachování pravidelnosti sítě (Klauda, 2016).

2.2 Použitá data

Při práci na této bakalářské práci byly použity dvě datové sady. Export z Nálezové databáze ochrany přírody poskytnutý Agenturou ochrany krajiny a přírody České republiky, na kterém byla testována metoda Marker Clustering a tvorba Heatmap pomocí JavaScriptových knihoven. Dále byla použita taxi data, která zdarma poskytuje New York City Taxi & Limousine Commission. Použité datové sady jsou podrobněji popsány v podkapitole 5.1 a 5.2.

2.3 Použité programy

V rámci bakalářské práce bylo otestováno několik JavaScriptových knihoven pro tvorbu interaktivních webových map a jejich pluginů. Pro tvorbu heatmap byly otestovány knihovny Leaflet 1.3.1 s pluginem Leaflet.heat 0.2 a knihovna OpenLayers 4.6.4, která

tvorbu heatmap nativně podporuje. Shlukování bodů bylo testováno za využití knihoven Leaflet 1.3.1 s pluginem Leaflet.markercluster 1.3.0, OpenLayers 4.6.4, Supercluster.js 3.0.2 (v kombinaci s knihovnou Leaflet), PruneCluster 2.1.0 (taktéž v kombinaci s knihovnou Leaflet) a Mapbox GL JS 0.44.1. Testování probíhalo v prohlížeči Google Chrome 65.0.3325.181 na lokálním webovém serveru Apache HTTP Server 2.4.29 (součást balíku XAMPP 7.2.3).

Dále byly použity open source nástroje GIS Tools for Hadoop, které slouží pro práci s prostorovými daty a umožňují provádět prostorové analýzy v Apache Hadoop frameworku. Testování těchto nástrojů probíhalo v prostředí Hortonworks Sandbox s HDP ve verzi 2.5 na platformě Microsoft Azure. Pro ovládání sandboxu byl využit SSH (Secure Shell) klient PuTTY ve verzi 0.70. Pro převod dat z formátu JSON na prvky a definování souřadnicového systému bylo využito nástrojů GIS Tools for Hadoop a programu ArcMap 10.5 z ArcGIS Desktop 10.5. Vizualizace dat probíhala v programu QGIS 2.18.18 a následně byl využit plugin QTiles pro export dlaždic.

Webové stránky a aplikace byly vytvořeny v textovém editoru Sublime Text 3. Pro tvorbu tabulek, sepsání textu práce a prezentaci výsledků byly využity programy Microsoft Excel, Microsoft Word a Microsoft PowerPoint z kancelářského balíku Microsoft Office 2016. Poster k bakalářské práci byl zpracován v programu Adobe Illustrator CC 2017.

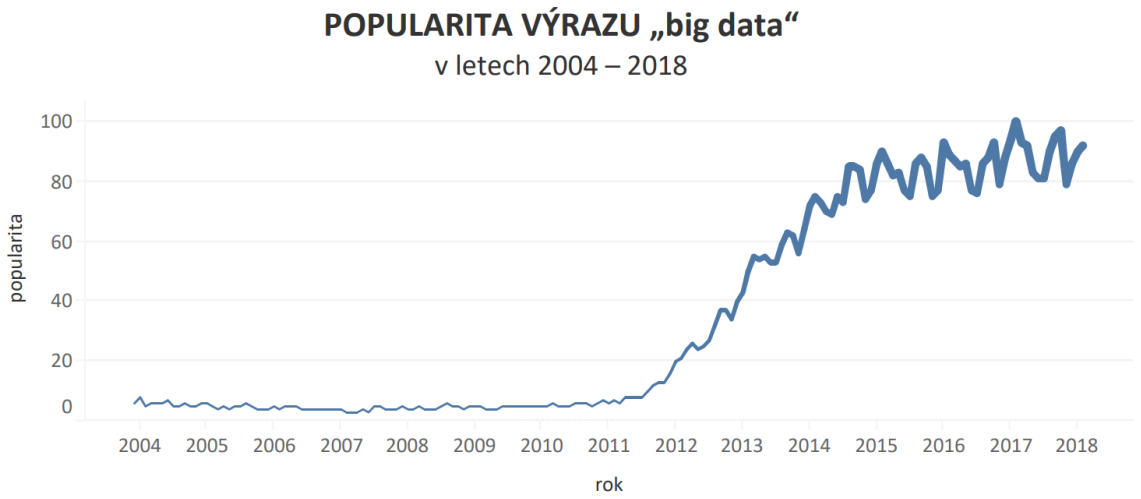
2.4 Postup zpracování

Jak již bylo zmíněno v úvodu této kapitoly, samotnému zpracování této bakalářské práce předcházelo studium odborné literatury, zabývající se technologiemi a postupy využívanými v oblasti Big Data. Následovalo vyhledání vhodných datových sad a nástrojů, se kterými bylo nutné se podrobně seznámit. Odlišnosti ve zpracování Big Data oproti konvenčnímu zpracování geodat byly demonstrovány na vybraném balíku nástrojů, kterým byl GIS Tools for Hadoop. Tento balík nástrojů byl použit v prostředí Hortonworks Sandbox s HDP na platformě Microsoft Azure. Pro zpracování dat pomocí GIS Tools for Hadoop byla zvolena metoda Spatial Binning. Dále bylo provedeno testování časové náročnosti zpracování dat pomocí této metody, konkrétně na agregaci dat do čtvercové sítě, a to na dvou výkonnostně odlišných virtuálních počítačích na platformě Microsoft Azure. Celý postup práce s nástroji GIS Tools for Hadoop je popsán v kapitole 7. Pro vizualizaci dat byly zvoleny JavaScriptové knihovny pro shlukování bodů a tvorbu heatmap, u kterých probíhalo testování rychlosti vykreslení na vytvořených vzorcích dat (předmět kapitoly 8). Na závěr byla sepsána textová část bakalářské práce, vytvořen informační poster a webové stránky, jejichž součástí jsou i vytvořené webové aplikace, které byly předmětem testování.

Obr. 1 Postup zpracování práce.

3 BIG DATA

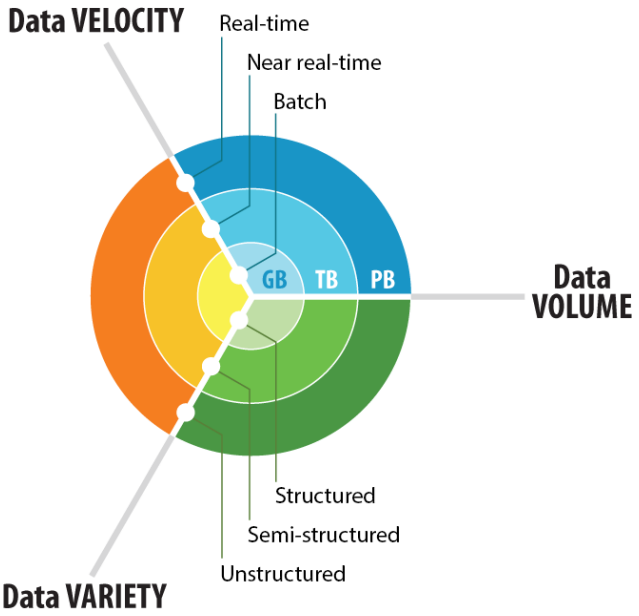
Big Data je termín, označující velmi objemné datové sady, které je obtížné ukládat, spravovat, sdílet, analyzovat a vizualizovat pomocí běžných nástrojů. Obrovský nárůst množství dat během posledních desetiletí je do značné míry následkem klesajících nákladů spojených s výpočetní technikou a informačními technologiemi. Se zvyšující se popularitou big dat (viz Obr. 2) se rozšiřují i technologie a možnosti, jak lze taková data zpracovávat.



Obr. 2 Popularita výrazu "big data" dle Google Trends[1] (zdroj: https://trends.google.com/trends).

Termín byl poprvé použit vědci NASA na 8. konferenci IEEE Visualization v roce 1997 v souvislosti s vizualizací dat, kdy se data stávají natolik objemná, že přesahují kapacitu paměti (Cox, Ellsworth, 1997). Nejčastěji se Big Data charakterizují pomocí takzvaných „3V“, která poprvé použil analytik společnosti Gartner Doug Laney. První výzkumná zpráva zabývající se těmito charakteristikami byla vypracována v roce 2000 a poté vydána v únoru roku 2001 pod názvem *3-D Data Management: Controlling Data Volume, Velocity and Variety*. Dle Laneyho (2012) se konkrétně jedná o objem (volume), rychlost nárůstu (velocity) a různorodost (variety). Tyto základní charakteristiky bývají některými autory a společnostmi často doplňovány o další. Podle IBM (2013) mají Big Data čtyři dimenze. Objem (volume), rychlost (velocity), různorodost (variety) a věrohodnost (veracity), vyjadřující nejistotu v datech. Firma Microsoft rozšířila původní charakteristiku o další dvě dimenze – proměnlivost (variability), která vyjadřuje na rozdíl od různorodosti počet proměnných v datové sadě a viditelnost (visibility) (Elbattah a kol. 2018). K těmto charakteristikám bývají často doplňovány i další. Někteří autoři doplňují hodnotu, kterou data představují pro firmu (value), dobu platnosti (validity), přechodnou dobu nutného ukládání dat (volatility) apod. (Holubová, 2015).

[1] Čísla představují relativní zájem ve vyhledávání vzhledem k nejvyššímu bodu grafu pro danou oblast a dobu. Hodnota 100 představuje nejvyšší popularitu výrazu. Hodnota 50 znamená, že měl výraz poloviční popularitu. Skóre 0 znamená, že pro výraz nebyl shromážděn dostatek dat (zdroj: https://trends.google.com/trends/).



Obr. 3 Charakteristika Big Data (zdroj: http://www.cousinsinfotech.com/big-data-analytics).

3.1 Velikost (Volume)

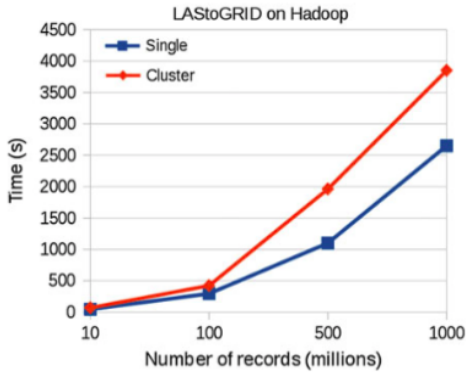
Jednou z charakteristik Big Data je velikost (volume), která může znamenat velikost dat jako takovou, případně velké množství záznamů, které dataset obsahuje. Podle odhadů společnosti IBM (2013) se denně vytvoří 2,5 kvintilionů bajtů dat a celkový objem dat bude do roku 2020 činit 40 zettabajtů. V dnešní době je běžné, že velké společnosti mají uložiště velké v řádech terabajtů či petabajtů. Příkladem takové společnosti může být například Facebook, který měl ke konci roku 2016 až 1,86 miliardy denně aktivních uživatelů, kteří denně sdílí až 500 TB dat. Nicméně Stephen Brobst, CTO (Chief Technology Officer) ve společnosti Teradata, v roce 2010 předpověděl, že během tří až pěti let nebudou největším zdrojem nestrukturovaných dat sociální sítě, nýbrž data získaná ze senzorů a senzorových sítí (Liang a Huang, 2014).

Velký zdroj 2D prostorových dat představují i letecké a satelitní snímky. V archivu USGS (United States Geological Survey) se k 1. 1. 2015 nacházelo 5 532 454 snímků z programu Landsat o celkové velikosti 4,134 PB (Wulder a kol. 2016) a denně získá NASA (National Aeronautics and Space Administration) přibližně 5 TB dat z dálkového průzkumu Země (Vatsavai a kol. 2012).

Dalším příkladem může být program Copernicus v rámci kterého družice Sentinel-1A a -1B pořídily ke konci roku 2016 celkem 0,77 PiB dat, Sentinel-2A 0,46 PiB dat a družice Sentinel-3A 10,4 TiB (Castriotta, 2017).

Rozsáhlým zdrojem 3D dat jsou data získaná pomocí technologie LIDAR (Light Detection And Ranging), sloužící k detekci objektů a měření vzdálenosti použitím laserového záření (Slovník VÚGTK, 2018). Uživatel tak může snadno získat milióny bodů ve zvolené zájmové oblasti. Tato bodová mračna lze poté využít k vytvoření 3D modelů skenovaných objektů. Právě zpracováním mračna bodů pomocí Apache Hadoop se zabýval Růžička a kol. (2017), kdy byla vytvořena MapReduce aplikace LAStoGRID, která byla následně otestována s odlišně velkými mračny bodů (10, 100, 500, 1000 miliónů bodů) za využití Apache Hadoop v konfiguraci *single-node* a clusteru (jeden *master* uzel a 3 *slave* uzly). Otestován byl i desktopový software, který pro tvorbu gridu běžně využívá katedra Geologického inženýrství na VŠB-TUO (Vysoká škola báňská – Technická univerzita Ostrava). Z testování vyplynulo, že desktopový software je schopen

zpracovat pouze vzorek o velikosti 10 miliónů bodů, a to za 537 sekund. Testy na vzorcích s vyšším počtem bodů skončily chybový hlášením. Využití Apache Hadoop přineslo značné snížení výpočetní doby pro tvorbu gridu. Ten byl vytvořen za 47 sekund v konfiguraci *single-node* a za 71 sekund při využití clusteru. Paradoxně konfigurace *single-node* dosahovala nižších časů zpracování než clusterové řešení. Možnou příčinou mohlo být použití jednoduchých výpočtů pro funkce *map* a *reduce* a distribucí dat mezi jednotlivými uzly (Růžička a kol. 2017).



Obr. 4 Doba potřebná pro tvorbu gridu na Apache Hadoop (Růžička a kol. 2017).

3.2 Rychlost (Velocity)

Rychlost (velocity) představuje, jak rychle jsou data generována a zpracována. Nestrukturovaná data rostou rychleji než data strukturovaná, a proto je nutné volit odlišné způsoby pro zpracování těchto dat. Data mohou být zpracovávána dávkově (např. pomocí MapReduce frameworku). Často je ale zpracovávat data v reálném čase. Pro takové potřeby existují další frameworky, mezi které můžeme zařadit například Apache Spark, Apache Storm či Apache Flink (Balusamy a kol. 2018).

3.3 Různorodost (Variety)

Podle Holubové a kol. (2015) hovoříme v oblasti Big Data o datech semi-strukturovaných (příkladem mohou být textové dokumenty CSV nebo data ve formátech XML či JSON) nebo zcela nestrukturovaných mezi které spadají například multimediální data. V případě dat uložených v klasických relačních databázích hovoříme potom o datech strukturovaných. Balusamy a kol. (2018) uvádí, že strukturovaná data tvoří pouze 5 % všech digitálních dat. Semi-strukturovaná data tvoří také 5 % a největší část (90 %) tvoří data nestrukturovaná.

4 DATOVÉ FORMÁTY

Tato kapitola popisuje vybrané datové formáty, se kterými je možné se setkat při zpracování Big Data v oblasti GIS a příbuzných oborech.

JSON

Formát JSON (JavaScript Object Notation) původně vznikl pro předávání dat mezi serverovou a klientskou částí webové aplikace. Jedná se o podmnožinu jazyka JavaScript, která dovoluje reprezentovat základní datové struktury a umožňuje jejich přímočaré použití v prohlížeči. Postupem času se však JSON stal rozšířeným datovým formátem a knihovny umožňující jeho použití existují pro všechny používané jazyky. Použití JSON v aplikaci je velice přímočaré, protože JSON lze snadno mapovat přímo na objekty daného jazyka (Holubová, 2015).

Formáty JSON mají své uplatnění především mezi webovými technologiemi. Oproti formátům odvozených z XML (GML, KML) mají kratší zápis, což je výhodné při přenosech v prostředí Internetu. Souřadnicový systém nelze definovat, a proto se předpokládá, že se jedná o WGS84 (World Geodetic System 1984). Data lze libovolným způsobem zanořovat a větvit (GISMentors, 2017).

GeoJSON

GeoJSON je výměnný geoprostorový formát založený na datovém formátu JSON. Definuje několik typů JSON objektů a způsob jejich kombinace, kterým reprezentují geografické prvky, jejich vlastnosti a prostorový rozsah. GeoJSON využívá souřadnicový systém WGS84 a jako jednotky desetinné stupně (RFC 7946 - The GeoJSON Format, 2016). GeoJSON je využíván u webových služeb pro svůj malý objem a jednoduchost. Je méně náročný na zpracování, což je vhodné zejména u webových prohlížečů (GISMentors, 2017).

Struktura formátu GeoJSON:

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "properties": {},
      "geometry": {
        "type": "Point",
        "coordinates": [17.265400886535645,49.593818757533676]
      }
    }
  ]
}
```

TopoJSON

TopoJSON je druhým formátem odvozeným z formátu JSON, který ale zatím nenabyl takové popularity jako GeoJSON. Hlavním úkolem formátu TopoJSON je minimalizace datového toku mezi webovým serverem a klientem. Formát je částečně ztrátový, neboť souřadnice bodů a lomových bodů jsou zapisovány v relativní poloze od daného počátku a v celých číslech (ztrácí se přesnost). K úspoře datové velikosti dochází také tím, že hranice polygonů jsou uloženy pro dvě sousedící plochy pouze jednou (formát je tedy topologický) (GISMentors, 2017).

Struktura formátu TopoJSON:

```
{
  "type":"Topology",
  "transform":{
    "scale": [1,1],
    "translate": [0,0]
  },
  "objects":{
    "two-squares":{
      "type": "GeometryCollection",
      "geometries":[
        {"type": "Polygon", "arcs":[[0,1]],"properties": {"name":
"Left_Polygon" }},
        {"type": "Polygon", "arcs":[[2,-1]],"properties": {"name":
"Right_Polygon" }}
      ]
    }
  },
  "arcs": [[[1,2],[0,-2]], [[1,0],[-1,0],[0,2],[1,0]], [[1,2],[1,0],[0,-
2],[-1,0]]]
}
```

CSV

Formát CSV (Comma-separated Values) je jednoduchý formát pro ukládání dat, která mají podobu tabulky. Každá řádka textového souboru obsahuje jeden záznam s několika položkami, které jsou oddělený oddělovačem. Nejčastěji se pro oddělení položek používá středník, čárka nebo tabulátor. První řádka souboru může obsahovat záhlaví sloupců (Holubová a kol. 2015).

Struktura formátu CSV:

```
Trip ID,Taxi ID,Trip Start Timestamp,Trip End Timestamp,Trip
Seconds,Trip Miles,Pickup Census Tract,Dropoff Census Tract,Pickup
Community Area,Dropoff Community Area,Fare,Tips,Tolls,Extras,Trip
Total,Payment Type,Company,Pickup Centroid Latitude,Pickup Centroid
Longitude,Dropoff Centroid Latitude,Dropoff Centroid Longitude
01e6,3999,12/01/2015 12:00:00 AM,12/01/2015 12:15:00
AM,120,0.3,17031081403,17031081600,8,8,$4.05,$0.00,$0.00,$0.00,$4.05,
Cash,Dispatch Taxi Affiliation,41.890922026,-87.618868355,41.892072635,-
87.628874157
```

XML

Značkovací jazyk XML (eXtensible Markup Languague) vznikl jako zjednodušení komplexního značkovacího jazyka SGML. Značkovací jazyky umožňují doplnit prostý text o strukturu elementů, která mu dá význam. V době vzniku XML byla zároveň velká poptávka po formátu, který by umožnil snadné propojení systémů a výměnu dat mezi nimi. XML šlo použít i k těmto účelům, a tak se stalo nejpoužívanějším formátem pro výměnu dat (Holubová a kol. 2015).

Na značkovacím jazyce XML jsou založeny i formáty KML (Keyhole Markup Languague) a GML (Geography Markup Languague). Formát KML byl původně vyvinut společností Google a umožňuje, na rozdíl od GML, použít pouze souřadnicový systém WGS84. Slouží především pro vizualizaci jednotlivých prvků geodat a od roku 2008 patří mezi standardy OGC (Open Geospatial Consortium). Formát GML je otevřeným OGC standardem pro přenos vektorových dat definovaný normou ISO 19136 (GISMentors, 2017).

Struktura formátu XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<current>
    <city id="2643741" name="City of London">
     <coord lon="-0.09" lat="51.51">
     <country>GB</country>
     <sun rise="2015-06-30T03:46:57" set="2015-06-30T20:21:12">
    </city>
    <temperature value="72.34" min="66.2" max="79.88"
unit="fahrenheit"/>
    <humidity value="43" unit="%">
    <pressure value="1020" unit="hPa">
    <wind>
     <speed value="7.78" name="Moderate breeze">
     <direction value="140" code="SE" name="SouthEast">
    </wind>
    <clouds value="0" name="clear sky">
    <visibility value="10000">
    <precipitation mode="no">
    <lastupdate value="2015-06-30T08:36:14">
</current>
```

5 ZDROJE DAT

Existuje celá řada datových sad vhodných pro zpracování technologiemi určenými pro zpracování Big Data. Můžeme se setkat s otevřenými daty, daty poskytovanými na základě smluv nebo daty získanými pomocí veřejně dostupných API či pomocí metod jako data mining a web crawling.

5.1 Nálezová databáze ochrany přírody

Nálezová databáze ochrany přírody (ND OP) je zdroj dat v oblasti druhové rozmanitosti České republiky a je poskytována Agenturou ochrany přírody a krajiny České republiky (AOPK ČR). Shrnuje veškerá dostupná data o rozšíření druhů na území ČR. Databáze je všeobecná, zaměřená na nálezy rostlin, živočichů i hub a lišejníků. Databáze je průběžně doplňována pomocí aplikace NDOP (dostupná na ndop.nature.cz), sloužící k editaci dat. Prohlížení dat pak umožňuje aplikace FiND (Filtr nálezových dat), která je dostupná pouze pro zaměstnance AOPK ČR, Ministerstva životního prostředí, případně na základě licenčních smluv. Aktuálně obsahuje databáze celkem přes 22 miliónu nálezů.

Pod pojmem nález se rozumí pozorování jedinců ve volné přírodě, ale také například nálezové údaje exemplářů ze sbírek a herbářů či každý publikovaný či nepublikovaný záznam o výskytu druhu na konkrétním území České republiky. Databáze se neomezuje na některou skupinu a data se sbírají o všech druzích. Primárně se jedná o ohrožené druhy, ale databáze obsahuje i značné množství záznamů o běžných druzích. Přesnost geografického a časového umístění se odvíjí od původního zdroje. U nově pořízených údajů je přesnost velmi vysoká, u historických údajů pak klesá.

Data jsou na základě smluv poskytována pro výzkumné účely včetně diplomových a disertačních prací. Data jsou poskytována ve formátech SHP (Shapefile) a CSV (Portál AOPK ČR, 2018). Záznamy v ND OP obsahují ID nálezu, taxon, autora nálezu, lokalitu, datum nálezu, početnost taxonu, souřadnice nálezu v souřadnicovém systému S-JTSK (Systém jednotné trigonometrické sítě katastrální) a poznámky.

Ukázka dat z Nálezové databáze ochrany přírody:

```
"ID_ND_NALEZ" "IDX_ND_AKCE" "DRUH" "AUTOR""LOKALITA""DATUM_OD"
"DATUM_DO" "ZDROJ" "POCET" "POP_POC" "REL_POC" "X" "Y" "IDX_ND_LOKAL"
"POZN1" "POZN2" "POZN3"
2395881 2990079 "Picea abies" "Tandler Libor" "m0174_251102_516_X9A"
"14.06.2003" "20.09.2003" "Libor Tandler. m0174: m0174. 2003." "" "" -
531068 -1111428 826685 "" "" ""
2011080 2877519 "Agrostis capillaris" "Kocurová Michaela"
"p0120_223305_113_L5.1" "04.07.2003" "13.08.2003" "Michaela Kocurová.
p0120: p0120. 2003." "" "" -813831 -1137483 714124 "" "" ""
```

5.2 NYC TLC Trip Data

Taxi data z města New York poskytuje NYC TLC (New York City Taxi & Limousine Commission) zdarma ve formátu CSV. Data jsou dostupná od roku 2009 pro žlutá taxi a od srpna roku 2013 pro zelená taxi (Boro taxi), která ale mají omezeno nabírání zákazníků na Manhattanu a zakázáno nabírání zákazníků na letišti JFK a letišti LaGuardia. Od roku 2015 jsou dostupná data pro tzv. FHV (For-Hire Vehicle), mezi které spadá například Uber, Lyft, Juno a Via. Záznamy ze žlutých a zelených taxi

obsahují časový údaj o nástupu a výstupu, přesné souřadnice nástupu a výstupu v souřadnicovém systému WGS84 (od července 2016 již nejsou poskytovány přesné souřadnice, ale pouze nástupní a výstupní taxi zóny), vzdálenost trasy, cenu jízdného, typ platby a počet cestujících. NYC TLC získává data od autorizovaných poskytovatelů v rámci programů Taxicab & Livery Passenger Enhancement (TPEP/LPEP). FHV data obsahují časový údaj nástupu, a ID lokality (NYC Taxi & Limousine Commission, 2018).

Obr. 5 Omezení zelených taxi (zdroj: http://www.nyc.gov).

Ukázka NYC TLC Taxi dat:

```
vendor_name,Trip_Pickup_DateTime,Trip_Dropoff_DateTime,Passenger_Count,T
rip_Distance,Start_Lon,Start_Lat,Rate_Code,store_and_forward,End_Lon,End
_Lat,Payment_Type,Fare_Amt,surcharge,mta_tax,Tip_Amt,Tolls_Amt,Total_Amt
VTS,2009-01-04 02:52:00,2009-01-04 03:02:00,1,2.63,-
73.991956,40.721567,,,-73.993803,40.695922,CASH,8.9,0.5,,0,0,9.4
VTS,2009-01-04 03:31:00,2009-01-04 03:38:00,3,4.55,-
73.982101,40.736289,,,-73.955849,40.768030,Credit,12.1,0.5,,2,0,14.6
DDS,2009-01-01 20:52:58,2009-01-01 21:14:00,1,5,-73.974266,40.790954,,,-
73.996557,40.731848,CREDIT,14.9,0.5,,3.049,0,18.4
```

5.3 Sociální sítě

Velkým zdrojem dat mohou být i sociální sítě, ze kterých lze získat data prostřednictvím tzv. API (Application Programming Interface). Příkladem takové sociální sítě může být například Twitter. Standartní verze Twitter Search API přístupná veřejnosti umožňuje vyhledávat ve tweetech, které jsou až 7 dní staré. Vyhledávání lze omezit pouze na určitý jazyk či zvolenou zájmovou oblast, ve které se mají tweety vyhledávat. Existuje také varianta Premium, vyžadující vývojářský účet a placená varianta Enterprise, která zahrnuje i technickou podporu. Obě tyto varianty pak umožňují vyhledávání tweetů již od roku 2006. Twitter Streaming API pak slouží k získávání tweetů v reálném čase. Veřejně poskytuje Twitter pouze malý vzorek dat a to nanejvýš 1 % tweetů v daném čase (Morstatter a kol. 2013). Dostupná je také varianta Enterprise, která tento limit navyšuje na 10 % (Twitter Developer Platform, 2018). Podle Croitoru a kol. (2014) se procento tweetů s přesně určenou polohou pomocí souřadnic pohybuje mezi 0,5 – 3 %. Popisná toponyma pak obsahuje přibližně 40-70 % tweetů.

Dalším příkladem může být sociální sít Foursquare, která ale neposkytuje přístup pomocí veřejného API k jednotlivým check-ins (přihlášení uživatele) na konkrétních místech (venues), nicméně ale existuje veřejně dostupná datová sada, obsahující

záznamy od dubna 2012 do září 2013, ve které je obsaženo celkem 33 278 683 check-ins od 266 909 uživatelů. Pro zachování soukromí uživatelů byla jejich uživatelská jména anonymizována. Datová sada je dostupná na následujícím odkazu: https://sites.google.com/site/yangdingqi/home/foursquare-dataset

5.4 Další datové zdroje

Mezi další datové zdroje můžeme zařadit klimatická data NOAA, která jsou dostupná hned z několika zdrojů. Konkrétně z datového portálu NOAA Earth Systems od společnosti IBM, AWS (Amazon Web Services) či Google Cloud Plaftorm. Dostupná jsou také taxi data pro město Chicago, a to od roku 2013. Tento dataset aktuálně tvoří celkem 112 860 054 záznamů a je dostupný zdarma ve formátech CSV, JSON či XML z oficiálního datového portálu města Chicago. Jako další zdroj lze považovat i tzv. VGI (Volunteered Geographic Information) data. Příkladem těchto dat mohou být například OpenStreetMap data, která mají aktuálně v nekomprimované podobě (formát XML) velikost přibližně 913 GB. V komprimované podobě (bzip2) mají tato data velikost 66,6 GB a ve formátu PBF pak 39,6 GB (OpenStreetMap Wiki, 2018).

6 HADOOP

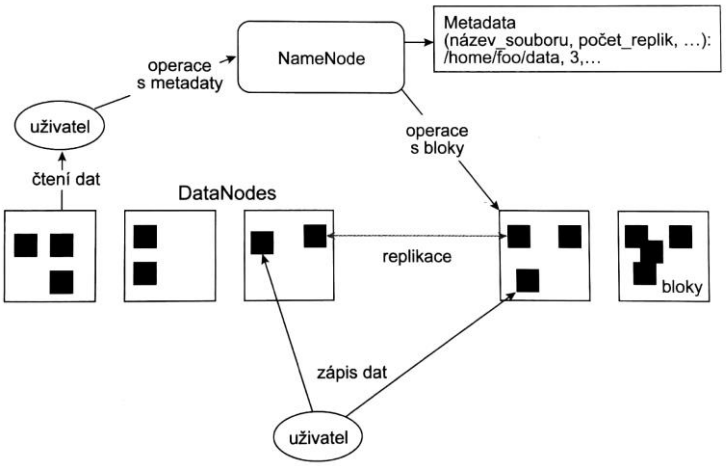
Jedná se o jedno z nejznámějších a nejefektivnějších řešení, jak distribuovaně zpracovávat a analyzovat velké objemy dat, která mohou být strukturovaná, částečně strukturovaná nebo nestrukturovaná. Dle Dumbilla (2013) se jedná řádově o desítky až stovky gigabitů či více. Apache Hadoop má své počátky v projektu Apache Nutch, což je open source webový vyhledávač, který je součástí Apache Lucene (knihovna pro fulltextové vyhledávání). Apache Hadoop je škálovatelný a vůči chybám tolerantní open source framework psaný v jazyce Java, jehož autorem je Doug Cutting a vývojem se zabývá společnost Apache Software Foundation. Jeho součástí je mnoho komponent, které mohou podle potřeby pracovat současně či samostatně. Mezi základní komponenty Apache Hadoop patří YARN (Yet Another Resource Negotiator), MapReduce a HDFS (Hadoop Distributed File System), které jsou popsány v následujících podkapitolách (Guller, 2015).

Apache Hadoop je navržen tak, aby běžel v clusteru počítačů využívajících běžně dostupný hardware, nicméně může běžet i na jednom počítači v *single-node* konfiguraci. Pro navýšení výpočetního výkonu a úložného prostoru se namísto vertikálního škálování (scaling up), kdy se výpočetní výkon zvyšuje použitím modernějšího a výkonnějšího hardwaru daného serveru, využívá škálování horizontální (scaling out), což znamená přidání dalšího uzlu do clusteru (Holubová a kol. 2015).

6.1 HDFS (Hadoop Distributed File System)

HDFS je škálovatelný distribuovaný souborový systém, který slouží pro ukládání souborů napříč clusterem na jednotlivých datových uzlech. Poskytuje rychlý přístup k velkým souborům, objemným datovým sadám a vyznačuje se vysokou tolerancí chyb. Soubor uložený v HDFS je rozdělen na několik bloků konfigurovatelné velikosti (výchozí 128 MB), které jsou uloženy na jednotlivých datových uzlech. (White, 2015). Díky distribuovanému uložení je čtení a zápis objemných dat mnohem rychlejší, než čtení a zápis velkých souborů z jednoho disku. S distribuovaným ukládáním se zvyšuje šance, že se soubor stane nedostupný v důsledku chyby datového uzlu. HDFS toto riziko snižuje tak, že jednotlivé bloky dat replikuje na více uzlech. Záměrně tedy dochází k redundanci dat, aby v případě selhání uzlu, byla data stále dostupná (Gulller, 2015).

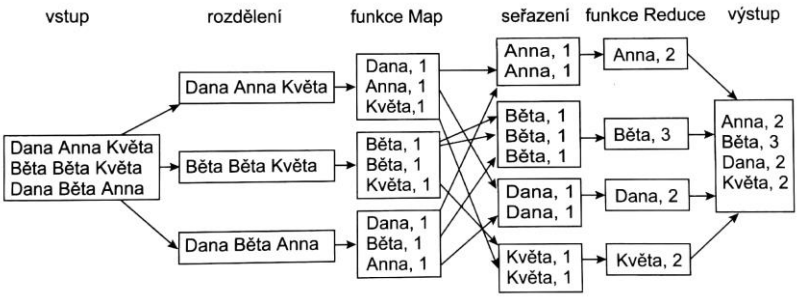
Základní prvky architektury HDFS tvoří uzly, které mohou být dvojího typu. Prvním je *namenode*, který spravuje jmenný prostor souborového systému a obsahuje veškerá metadata (názvy souborů, uživatelská oprávnění a místo uložení datových bloků). Bez *namenode* by distribuovaný souborový systém nemohl fungovat, jelikož by bez něj nebylo možné rekonstruovat data z bloků uložených na datových uzlech (*datanode*). Aby nedošlo ke ztrátě dat, poskytuje Hadoop dvě možnosti zálohy. Zálohovat lze samotná metadata na lokální disk případně prostřednictvím protokolu NFS (Network File Systém) na disk vzdáleného serveru. Druhou možností je vytvoření sekundárního *namenode* běžícího na samostatném počítači, který uchovává obraz jmenného prostoru a periodicky jej aktualizuje o provedené změny uložené v logu (White, 2015).



Obr. 6 Architektura HDFS (Holubová a kol. 2015).

6.2 MapReduce

MapReduce je programovací model představený společností Google v roce 2004 (Dean a Ghemawat, 2004), který umožňuje paralelní zpracování objemných datových sad v clusteru. Umožňuje programátorům bez předchozích zkušeností s psaním paralelních aplikací psát aplikace, které jsou schopny běžet na clusteru běžného hardwaru. MapReduce automaticky plánuje spuštění aplikace na clusteru, spravuje vyvažování zátěže na jednotlivých uzlech, chyby uzlů a komunikaci mezi nimi. Základním stavebním prvkem MapReduce frameworku jsou funkce *map* a *reduce*. Vstupní data jsou po načtení rozdělena na menší části podle páru *klíč-hodnota*, na kterých následně proběhne funkce *map*, jejímž výstupem jsou přechodné páry *klíč-hodnota*. Tyto páry jsou poté rozděleny a seřazeny. Seřazené páry vstupují do funkce *reduce*, která agreguje všechny stejné páry do jednoho výsledného páru (Guller, 2015). Proces mapovací a redukční fáze je zobrazen na Obr. 7.



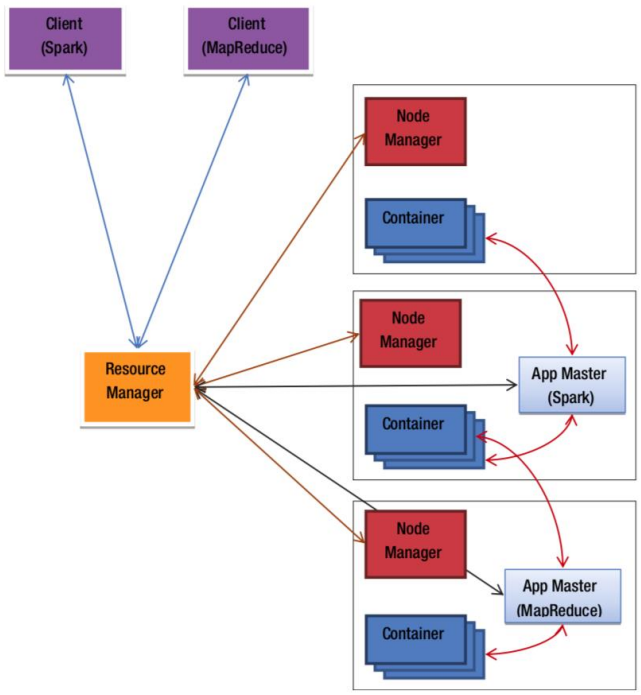
Obr. 7 Průběh mapovací a redukční fáze (Holubová a kol. 2015).

6.3 YARN

Apache YARN (Yet Another Resource Negotiator) je systémem řízení zdrojů v Hadoop clusteru. YARN byl poprvé představen v Hadoop verzi 2 pro vylepšení implementace MapReduce. Obecně ale slouží i pro podporu ostatních paradigmat v oblasti distrubovaného zpracování dat. První verze Apache Hadoop podporovala pouze jeden framework pro distribuované zpracování dat – MapReduce (označován také jako MapReduce 1), který pomocí uzlu *JobTracker (master)* zajišťoval plánování úloh (přiřazování úloh *TaskTracker (slave)* uzlům), rozdělování dostupných prostředků

v clusteru a monitorování průběhu výpočtů. Uzly *TaskTrackers* pak sloužily pro provádění výpočtů, jejichž výsledek byl předán zpět uzlu *JobTracker*, který ukládál výsledek každé provedené úlohy. V případě chybného zpracování úlohy, mohl *JobTracker* úlohu přeplánovat na jiný *TaskTracker* uzel (White, 2015). S uvedením YARN jsou funkce uzlu *JobTracker* rozděleny mezi dvě samostatné entity – *ResourceManager* a *ApplicationMaster* (viz Obr. 8).

Centrální ResourceManager se dále dělí na dvě komponenty – *Scheduler a ApplicationsManager*. *Scheduler* je komponenta, zajišťující plánování a přidělování prostředků entitě *ApplicationMaster*, představující instanci použitého frameworku pro zpracování dat (MapReduce 2, Spark apod.), která odpovídá za vyžádání potřebných prostředků, sledování stavu a monitorování průběhu aplikace. Druhá komponenta *ApplicationsManager* přijímá úlohy od klientské aplikace a přiřazuje jim kontejner, jež představuje přidělené prostředky a běží v něm právě jedna spuštěná úloha. *TaskTracker* byl nahrazen komponentou *NodeManager*, což je služba, která existuje samostatně na každém z uzlů. Odpovídá za přiřazené kontejnery, využití prostředků (CPU, paměť, disk, síť) a hlášení těchto informací zpět centrální entitě *ResourceManager* (Apache Software Foundation, 2018).



Obr. 8 Architektura Apache YARN (Guller, 2015).

6.4 Apache Hive

Tento framework vznikl z důvodu potřeby zpracovávat objemná data, denně produkovaná společností Facebook, která byla ukládána v HDFS (White, 2015). Dnes je Apache Hive využíván mnoha podniky jako škálovatelná platforma pro zpracování dat. Jedná se o datový sklad, který poskytuje jazyk HiveQL (Hive Query Languague) pro zpracování a analýzu dat uložených v HDFS. Pomocí Apache Hive lze přistupovat i k datům, která jsou uložená v jiných systémech kompatibilních s Apache Hadoop (např. Apache Cassandra, Apache HBase apod.) (Guller, 2015).

Toto speciální datové úložiště, pracuje převážně s velkým objemem dat, která mohou být integrována z různých zdrojů. Pomocí Apache Hive je možné datům uložených v HDFS definovat a přiřadit strukturu, která je podobná relačnímu modelu. Poté je možné se nad nimi dotazovat pomocí jazyka HiveQL, vycházejícího z jazyka SQL. Dotazy napsané v HiveQL jsou pak kompilátorem přeloženy na sérii MapReduce úloh, které jsou spuštěny nad MapReduce frameworkem. Jazyk HiveQL se dělí na dvě části - DDL (Data Definition Languague) pro definování dat (např. příkaz `CREATE TABLE`) a DML (Data Manipulation Languague) pro manipulaci s daty (např. příkaz `LOAD DATA`, který načte data z HDFS do vytvořené tabulky). Pro dotazování se v HiveQL využívá příkazu `SELECT`. Výsledky HiveQL dotazů je lze poté vložit do vytvořených tabulek (Holubová a kol. 2015). Apache Hive umožňuje také vytvářet uživatelsky definované funkce (UDF) či uživatelsky definované agregační funkce (UDAF) pro zpracování komplexních úloh a rozšíření funkcionality.

6.5 Hadoop distribuce

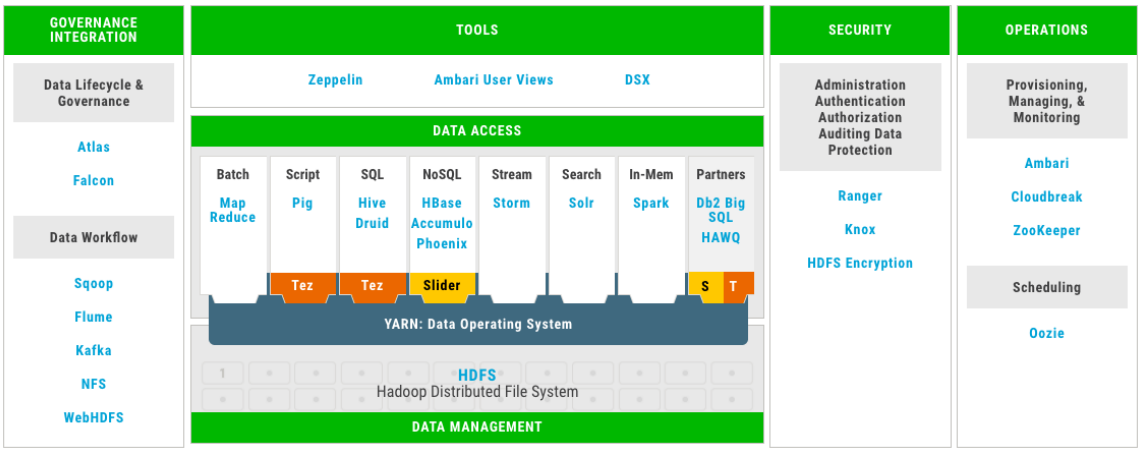
Apache Hadoop jako takový je open-source software poskytovaný zdarma. Od jeho uvedení v roce 2011 ale vznikla celá řada komerčních Hadoop distribucí pro specifické potřeby různých firem, které kombinují Apache Hadoop s ostatními Apache projekty.

Cloudera CDH

Jednou z nejznámějších a nejrozšířenějších distribucí Apache Hadoop představuje americká společnost Cloudera. Společnost byla založena v roce 2008 a jejími zakladateli byli Christophe Bisciglia ze společnosti Google, Amr Awadallah (Yahoo!), Mike Olson (Oracle) a Jeff Hammerbacher (Facebook). V roce 2009 se do společnosti připojil i Doug Cutting, autor Apache Hadoop. Cloudera byla první společností, která nabídla Hadoop jako kompletní balík a nadále patří mezi nejpoužívanější distribuce. Její distribuce CDH (Cloudera's Distribution Including Apache Hadoop) je open-source pod licencí Apache 2.0 a je poskytována zdarma. Aktuální verze nese označení 5.13. Proprietárním produktem je balík Cloudera Enterprise dostupný v pěti různých verzích, které se odlišují svými funkcemi. Hlavním nástrojem je Cloudera Manager, který slouží pro nastavení a monitorování clusteru (Cloudera, 2018).

Hortonworks Data Platform (HDP)

Mezi další společnosti poskytující distribuce Apache Hadoop patří Hortonworks. Hortonworks je americká společnost, která vznikla v roce 2011. Byla založena 24 inženýry z Hadoop týmu společnosti Yahoo!. Její distribuce Apache Hadoop nese název Hortonworks Data Platform (HDP) a ze všech distribucí má nejvíce přispěvatelů do zdrojového kódu. Společnost poskytuje placenou podporu a služby, ale neposkytuje žádný proprietární software a je zaměřená jen na open-source. První verze vyšla v roce 2012 a nynější aktuální verze této distribuce je 2.6.4. Monitorování a správu clusteru zajišťuje projekt Apache Amabari, který poskytuje jednoduché a přehledné webové rozhraní (Hortonworks, 2018).



Obr. 9 Ekosystém Hortonworks Data Platform (zdroj: https://hortonworks.com/products/data-platforms/hdp).

MapR Converged Data Platform

Třetí hlavní distribucí je MapR Converged Data Platform od americké společnosti MapR Technologies, Inc., která byla založena v roce 2009. Distribuce je poskytována ve dvou verzích. Community Edition je poskytována zdarma a placená verze nese název Enterprise Edition. Distribuce MapR Converged Data Platform je známá především svým distribuovaným souborovým systémem MapR-FS. MapR-FS je POSIX distribuovaný souborový systém, který podporuje HDFS API a rychlý NFS přístup. Od distribuovaného souborového systému HDFS se odlišuje zejména tím, že v něm neexistuje *namenode* uzel, který v HDFS spravuje uložení dat na jednotlivých datových uzlech a metadata. MapR-FS tento koncept nevyužívá a veškeré tyto informace spravují samotné datové uzly. Tímto se tento souborový systém stává tolerantnější vůči chybám a spolehlivější než HDFS. Druhou odlišností je, že MapR-FS je napsán v nativním kódu kdežto HDFS je psáno v jazyce Java (Kim, 2016). Distribuce MapR Converged Data Platform používá vlastní NoSQL (Not only SQL) databázi MapR-DB, která je podle analýzy společnosti ESG (Enterprise Strategy Group) v průměru dvaapůlkrát výkonnější než NoSQL databáze Cassandra a dokonce pětinásobně převyšuje výkon NoSQL databáze HBase (Leone a Amato, 2017).

HDInsight

Mezi cloudové implementace Hadoop můžeme zařadit službu HDInsight, která je dostupná na cloudové platformě Azure společnosti Microsoft. Společnost Microsoft vytvořila službu HDInsight v úzké spolupráci s Hortonworks na jejichž HDP distribuci je tato služba postavena. Při tvorbě clusteru lze volit mezi různými typy – Hadoop, Spark, Storm, Kafka apod. Službu HDInsight bylo možné provozovat i na platformě Windows Server 2012 R2, nicméně od verze HDInsight 3.4 byla podpora této platformy ukončena a v budoucích verzích lze vytvářet pouze clustery postavené na operačním systému Linux. Aktuální verze HDInsight 3.6 pracuje konkrétně na linuxové distribuci Ubuntu ve verzi 16.0.4 LTS. Cena za službu HDInsight se liší podle zvolené konfigurace uzlů a jejich počtu. Limit velikosti clusteru (počet uzlů) se odvíjí podle předplatného pro daný Azure účet a lze jej libovolně měnit (Azure HDInsight Documentation, 2018).

Elastic MapReduce (EMR)

Elastic MapReduce je služba společnosti Amazon představená v dubnu 2009. Jedná se o cloudovou Hadoop distribuci, která běží na infrastruktuře Amazon Elastic Compute Cloud (Amazon EC2), která je snadno škálovatelná a umožňuje měnit počet uzlů obdobně jako HDInsight. Na EMR lze také spustit další frameworky pro distribuované zpracování dat mezi které patří například Apache Spark či Apache Flink. Zajištěna je i interakce s daty uloženými v ostatních AWS (Amazon Web Services) úložištích Amazon S3 (Simple Storage Service) a Amazon DynamoDB. Využitím Elastic MapReduce distribuce uživateli odpadá nutnost konfigurovat uzly, vybraný framework pro distribuované zpracování dat a Hadoop cluster (Amazon EMR, 2018).

7 TESTOVÁNÍ GIS TOOLS FOR HADOOP

GIS Tools for Hadoop je open source sada nástrojů, která je schopna pracovat s Big daty obsahujícími lokaci a umožňuje provádět prostorové analýzy za využití distribuovaného zpracování v Hadoop frameworku. Samotná sada nástrojů se skládá z projektů Esri Geometry API for Java, Spatial Framework for Hadoop a Geoprocessing Tools for Hadoop.

Esri Geometry API for Java

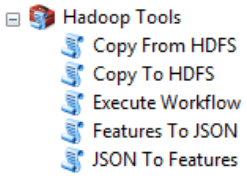
Tato knihovna umožňuje podporu geometrických prvků (body, linie a polygony), prostorových operací (např. průnik, buffer) a prostorového indexování. Pomocí této knihovny lze také vytvářet vlastní MapReduce úlohy v jazyce Java pro další analýzy na prostorových datech.

Spatial Framework for Hadoop

Tato knihovna obsahuje uživatelsky definované funkce, které rozšiřují možnosti datového skladu Apache Hive a jsou postaveny na knihovně Esri Geometry API for Java. Při spuštění této knihovny v Apache Hive, lze vytvářet dotazy za pomoci dotazovacího jazyka HiveQL, který je velmi podobný jazyku SQL (Structured Query Language), namísto psaní samotných MapReduce úloh.

Geoprocessing Tools for Hadoop

Jedná se o toolbox, který obsahuje sadu nástrojů pro propojení ArcMap a Hadoop. Nástrojem *Features to JSON* lze převést prvky do formátu JSON, které lze následně pomocí nástroje *Copy to HDFS*, zkopírovat do distribuovaného souborového systému HDFS. Výsledek analýzy v Hadoop je poté možné pomocí nástroje *Copy from HDFS* vyexportovat ve formátu JSON a nástrojem *JSON to Features* převést vyexportované JSON soubory na prvky pro další zpracování v ArcMap. Posledním nástrojem je *Execute workflow*, který umožňuje spouštět úlohy naplánované v Apache Oozie pro systém Hadoop (Esri, 2015).



Obr. 10 Geoprocessing Tools for Hadoop toolbox.

Pro testování GIS Tools for Hadoop je doporuřen virtuální stroj s Hadoop prostředím – konkrétně se jedná o Hortonworks Sandbox ve virtuálním prostředí VirtualBox. Pro potřeby této bakalářské práce bylo pro nasazení HDP (Hortonworks Data Platform) využito cloudové výpočetní platformy Microsoft Azure. Prvním krokem je vyhledání Hortonworks Sandbox with HDP 2.5 na Microsoft Azure Marketplace. Následně se podle průvodce provede konfigurace virtuálního počítače. Zvolíme název virtuálního počítače, uživatelské jméno, heslo (případně veřejný SSH klíč). Po nasazení HDP Sandboxu bylo nutné povolit příchozí síťová připojení – konkrétně se jedná o porty 8888, 8080 (Ambari) a 4200 (vestavěný SSH klient). Povolení těchto portů provedeme v nastavení síťového rozhraní (viz Obr. 11).

Obr. 11 Nastavení síťového rozhraní.

Po nakonfigurování a úspěšném nasazení je HDP Sandbox dostupný na IP adrese `40.68.254.xx:8888`. Pro správu a sledování clusteru lze využít webové rozhraní Ambari dostupné na `40.68.254.xx:8080`. Dostupný je také vestavěný SSH klient na adrese `40.68.254.xx:4200`. Pro přístup k HDP Sandboxu lze využít i jiných klientu jako např. PuTTY, který byl využit i v této bakalářské práci. V tomto případě je ale nutné pro připojení použít port 22. Pro potřeby testování nástroju GIS Tools for Hadoop je v rámci této bakalářské práce HDP v konfiguraci *single-node*.

Nejprve je nutné se přihlásit do vytvořeného virtuálního počítače na přidělené IP adrese `40.68.254.xx` pomocí přihlašovacích údajů zadaných při konfiguraci a teprve poté do HDP Sandboxu. Připojení do HDP Sandboxu je uskutečněno příkazem `ssh root@172.17.0.2 -p 22`. Ve výchozím nastavení je uživatelské jméno a heslo pro přihlášení do HDP Sandboxu ve tvaru root/hadoop. Po prvotním přihlášení je uživatel z bezpečnostních důvodů vyzván k zadání nového hesla.

Prvním krokem bylo vytvoření adresáře, do kterého byl následně naklonován GIS Tools for Hadoop toolkit. Adresář vytvoříme pomocí příkazu `mkdir esri-git`. Zdali se adresář opravdu vytvořil, lze ověřit příkazem `ls`, který vypíše seznam složek a souborů v aktuálním umístění. Dalším krokem je přesun do nově vytvořeného adresáře pomocí příkazu `cd esri-git` a naklonování repozitáře `gis-tools-for-hadoop` příkazem `git clone git@192.30.253.112:Esri/gis-tools-for-hadoop.git`.

7.1 Agregace bodů v polygonu

Nejdříve byl vytvořen pracovní adresář `earthquakes` v HDFS příkazem `hadoop fs -mkdir earthquake`, do kterého byla následně nakopírována testovací data pomocí dvou následujících příkazů.

```
hadoop fs -put gis-tools-for-hadoop/samples/data/counties-data
earthquake
hadoop fs -put gis-tools-for-hadoop/samples/data/earthquake-data
earthquake
```

Poté byl spuštěn Hive CLI (Command Line Interface) pomocí `hive` a importovány externí java knihovny (Esri Geometry API a Spatial Framework for Hadoop) pro práci s prostorovými daty.

```
add jar
  ${env:HOME}/esri-git/gis-tools-for-hadoop/samples/lib/esri-geometry-
api-2.0.0.jar
  ${env:HOME}/esri-git/gis-tools-for-hadoop/samples/lib/spatial-sdk-
hive-2.0.0.jar
  ${env:HOME}/esri-git/gis-tools-for-hadoop/samples/lib/spatial-sdk-
json-2.0.0.jar;
```

Následně bylo nutné definovat uživatelské funkce *ST_Point(x, y)* pro vytvoření bodu ze souřadnic x, y) a *ST_Contains(geometry1, geomtery2)*, která vrátí kladnou hodnotu pokud *geomtery1* obsahuje *geomtery2*.

```
create temporary function ST_Point as'com.esri.hadoop.hive.ST_Point';
create temporary function ST_Contains as
'com.esri.hadoop.hive.ST_Contains';
```

Dalším krokem bylo vytvoření tabulek pro data o zemětřesení (`earthquakes`), okresy v Kalifornii (`counties`), definování datového typu u jednotlivých atributů, formátu vstupních dat a oddělovače. V případě dat o zemětřesení se jednalo o textový soubor ve formátu CSV a okresy Kalifornie byly uloženy ve formátu JSON.

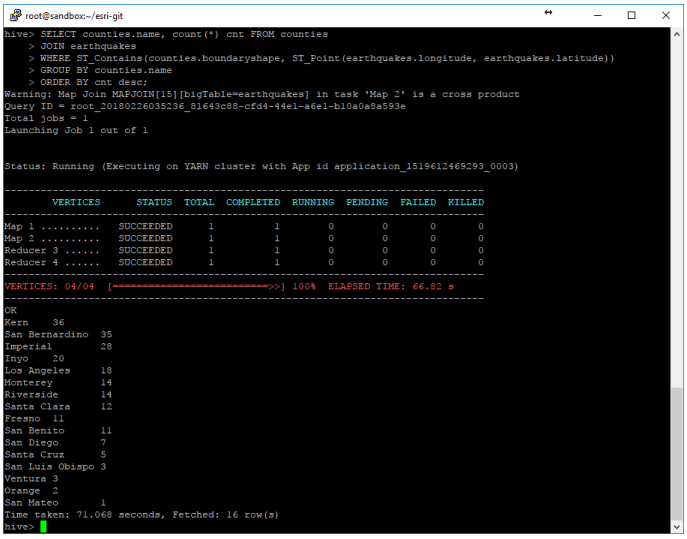
```
CREATE TABLE earthquakes (earthquake_date STRING, latitude DOUBLE,
longitude DOUBLE, depth DOUBLE, magnitude DOUBLE, magtype string,
mbstations string, gap string, distance string, rms string, source
string, eventid string)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE;

CREATE TABLE counties (Area string, Perimeter string, State string,
County string, Name string, BoundaryShape binary)
ROW FORMAT SERDE 'com.esri.hadoop.hive.serde.EsriJsonSerDe'
STORED AS INPUTFORMAT 'com.esri.json.hadoop.EnclosedEsriJsonInputFormat'
OUTPUTFORMAT
'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat';
```

Dále bylo provedeno načtení dat (`earthquakes.csv` a `california-counties.json`) do jednotlivých tabulek (`earthquakes` a `counties`) pomocí příkazu `LOAD DATA`.

```
LOAD DATA INPATH 'earthquake/earthquake-data/earthquakes.csv' OVERWRITE
INTO TABLE earthquakes;
LOAD DATA INPATH 'earthquake/counties-data/california-counties.json'
OVERWRITE INTO TABLE counties;
```

Následně byla spuštěna prostorová analýza, jejímž výsledkem je počet zemětřesení v jednotlivých okresech Kalifornie (viz Obr. 12).

```
SELECT counties.name, count(*) cnt FROM counties
JOIN earthquakes
WHERE ST_Contains(counties.boundaryshape,
ST_Point(earthquakes.longitude, earthquakes.latitude))
GROUP BY counties.name
ORDER BY cnt desc;
```



Obr. 12 Výsledek prostorové analýzy v Hive.

7.2 Agregace bodů do čtvercové sítě

Nejdříve bylo nutné zkopírovat data z lokálního úložiště do HDP Sandboxu. Toto lze provést několika způsoby. Pro přenos lze využít nástroj PSCP (PuTTY Secure Copy), který je součástí software PuTTY. Alternativou může být software WinSCP se kterým je snazší manipulace, jelikož se jedná o plnohodnotného SFTP, FTP a SCP klienta s grafickým uživatelským rozhraním. Na platformách macOS a Linux lze využít terminál a pro zkopírování dat použít příkaz (vyžadováno OpenSSH 7.3).

```
scp -oProxyJump=hdp@40.68.254.xx yellow_tripdata_2010-01.csv
root@172.17.0.2:/root
```

Po zkopírování dat do HDP Sandboxu bylo nutné vytvořit adresář v HDFS, kde budou data uložena. To bylo provedeno příkazem `hadoop fs -mkdir taxi-data`.

Následoval přesun dat do HDFS pomocí příkazu `hadoop fs -put taxi-data/trip_data_1.csv taxi-data` a import externích knihoven Esri Geometry API a Spatial Framework for Hadoop.

```
add jar
  ${env:HOME}/esri-git/gis-tools-for-hadoop/samples/lib/esri-geometry-
api-2.0.0.jar
  ${env:HOME}/esri-git/gis-tools-for-hadoop/samples/lib/spatial-sdk-
hive-2.0.0.jar
  ${env:HOME}/esri-git/gis-tools-for-hadoop/samples/lib/spatial-sdk-
json-2.0.0.jar;
```

V dalším kroku bylo nutné vytvořit uživatelské funkce *ST_Bin* – vrátí hodnotu bin id, ve kterém je daný bod obsažen, *ST_Point(x, y)* – vytvoření bodu ze souřadnic x, y a *ST_BinEnvelope(binsize, point)* – vrátí obálku daného bodu.

```
create temporary function ST_Bin as 'com.esri.hadoop.hive.ST_Bin';
create temporary function ST_Point as 'com.esri.hadoop.hive.ST_Point';
create temporary function ST_BinEnvelope as
'com.esri.hadoop.hive.ST_BinEnvelope';
```

Následovalo vytvoření tabulky `yellow_tripdata_2010_01`, ve které byly definovány datové typy jednotlivých polí, oddělovač, typ vstupního souboru a informace o tom, že první řádek má být přeskočen, jelikož se jedná o záhlaví.

```
CREATE EXTERNAL TABLE yellow_tripdata_2010_01(vendor_name STRING,
Trip_Pickup_DateTime STRING, Trip_Dropoff_DateTime STRING,
Passenger_Count DOUBLE, Trip_Distance DOUBLE, Start_Lon DOUBLE,
Start_Lat DOUBLE, Rate_Code STRING, store_and_forward STRING, End_Lon
DOUBLE, End_Lat DOUBLE, Payment_Type STRING, Fare_Amt DOUBLE, surcharge
DOUBLE, mta_tax DOUBLE, Tip_Amt DOUBLE, Tolls_Amt DOUBLE, Total_Amt
DOUBLE)
ROW FORMAT delimited fields terminated by ',' STORED AS textfile
tblproperties ("skip.header.line.count"="1");
```

Načtení dat v CSV souboru do tabulky `yellow_tripdata_2010_01` proběhlo příkazem

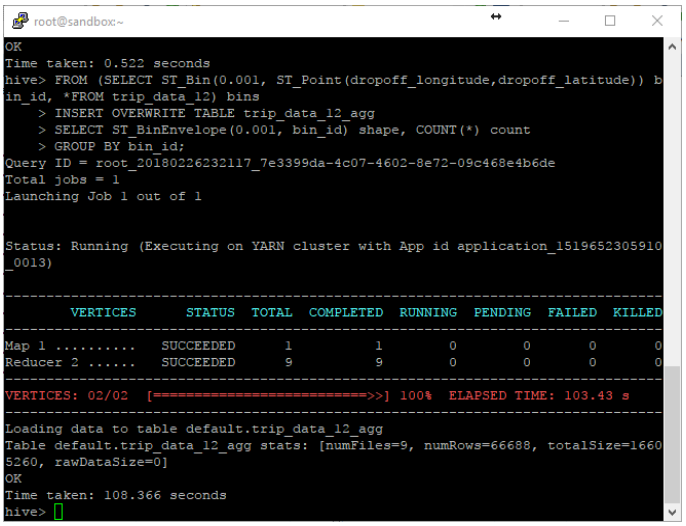
```
LOAD DATA INPATH 'taxi-data/yellow_tripdata_2010-01.csv' OVERWRITE INTO
TABLE yellow_tripdata_2010_01;
```

Dále byla vytvořena tabulka `yellow_tripdata_2010_01_agg_dropoffs` pro agregovaná data (výstupní místa v měsíci lednu roku 2010) a definování výstupního typu souboru (v tomto případě se jedná o JSON).

```
CREATE TABLE yellow_tripdata_2010_01_agg_dropoffs(area BINARY, count
DOUBLE)
ROW FORMAT SERDE 'com.esri.hadoop.hive.serde.EsriJsonSerDe'
STORED AS INPUTFORMAT
'com.esri.json.hadoop.UnenclosedEsriJsonInputFormat'
OUTPUTFORMAT
'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat';
```

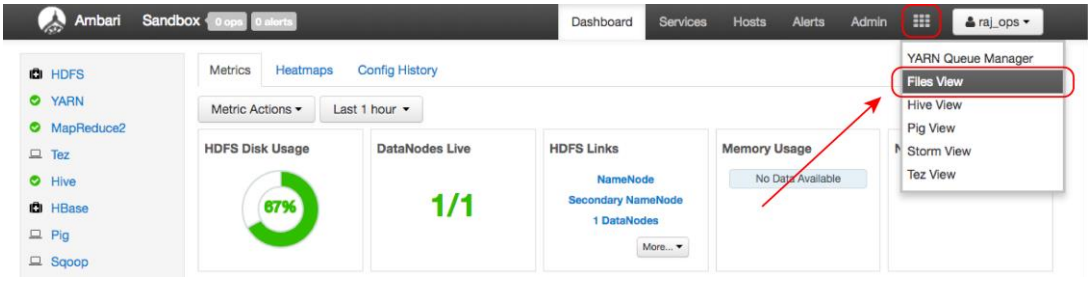
Poté následovalo definování dotazu pro agregaci (podrobnost výsledné čtvercové sítě v desetinných stupních, souřadnice bodů výstupních míst a tabulka, do které má být výsledek uložen) a jeho provedení (Obr. 13).

```
FROM (SELECT ST_Bin(0.001, ST_Point(End_Lon,End_Lat)) bin_id, *FROM
yellow_tripdata_2010_01) bins
INSERT OVERWRITE TABLE yellow_tripdata_2010_01_agg_dropoffs
SELECT ST_BinEnvelope(0.001, bin_id) shape, COUNT(*) count
GROUP BY bin_id;
```



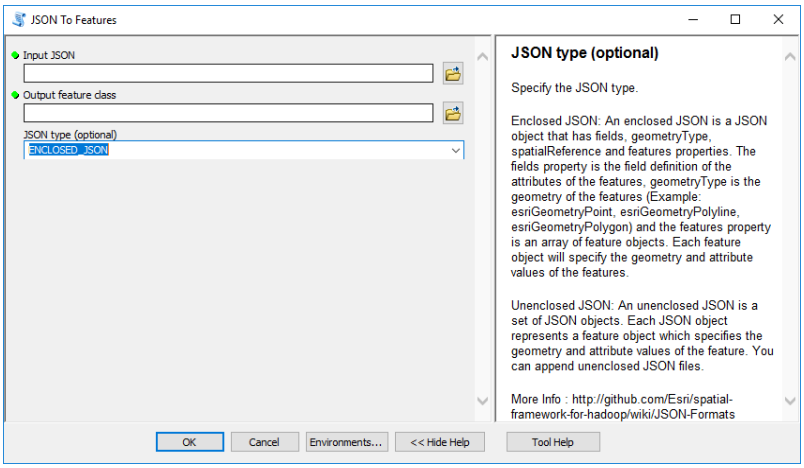
Obr. 13 Výsledek agregace v Hive.

Po dokončení agregace bylo nutné výsledná data exportovat z HDFS. V případě, že by byl HDP provozován na lokálním počítači (serveru), bylo by možné pro export použít nástroj *Copy from HDFS*, který je součástí Geoprocessing Tools for Hadoop toolboxu. Jelikož byl ale virtuální počítač s HDP Sandbox hostován na vzdáleném serveru platformy Microsoft Azure a nástroj *Copy from HDFS* neumožňuje připojení přes proxy, bylo nutné pro export dat z HDFS využít webového rozhraní pro správu clusteru Ambari. Po přihlášení do webového rozhraní Ambari pomocí přihlašovacího jména a hesla (raj_ops/raj_ops) bylo nutné přejít do správce souborů, který umožňuje prohlížení souborů uložených v HDFS. Tento správce souboru se v Ambari nachází pod záložkou „Files View“ (Obr. 14). Výsledné soubory se nachází v umístění `apps/hive/warehouse`.



Obr. 14 Ambari Files View.

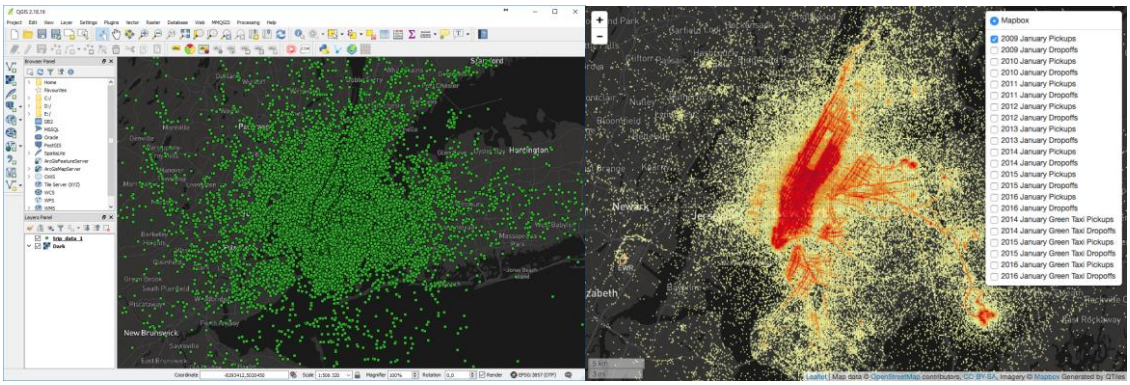
Počet výsledných soubor se odvíjí od počtu redukčních funkcí, které byly pro zpracování dat použity. V příkladu, který je uveden na Obr. 13 se bude tedy jednat o 9 souborů. Ty je možné stáhnout buď jednotlivě nebo najednou v komprimovaném formátu ZIP. Po stažení a rozbalení těchto souborů do zvoleného adresáře je nutné souborům doplnit příponu *.json aby mohly být správně načteny a dále zpracovány v programu ArcMap. Pro hromadné přejmenování všech souborů byl použit program Total Commander a jeho funkce hromadné přejmenování. Následně bylo nutné pomocí nástroje *JSON to Features* převést JSON soubory na prvky.



Obr. 15 Nástroj JSON To Features.

Pro usnadnění práce je možné vytvořit toolbox v prostředí ModelBuilder, který tento převod zcela automatizuje, nicméně při zpracování pomocí toolboxu neprobíhal python skript pro převod dat korektně, a tak bylo využito pouze dávkového zpracování souborů. Převedené JSON soubory na prvky byly poté sloučeny nástrojem *Merge*. Výsledné vrstvy dat nemají definovaný žádný souřadnicový systém, a tak jej bylo nutné pomocí nástroje *Define Projection* definovat. Nastaven byl souřadnicový systém WGS84, ve kterém jsou data od NYC TLC poskytována. Tímto způsobem byly vytvořeny vrstvy agregovaných dat pro měsíc leden v letech 2009–2016 pro žlutá taxi a v letech 2014–2016 pro zelená Boro taxi. Data jsou dostupná i za rok 2017 nicméně, od července roku 2016 NYC TLC již neposkytuje přesné GPS souřadnice nástupů a výstupů z taxi.

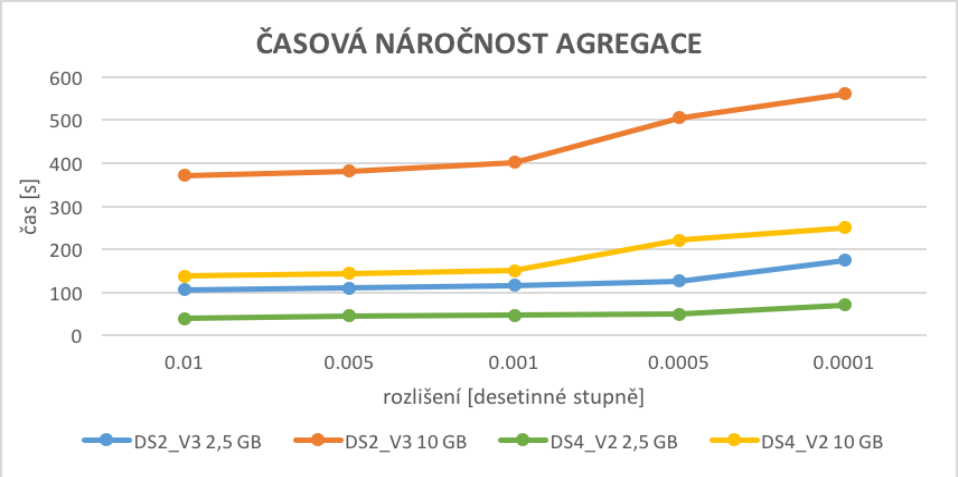
Následná vizualizace dat poté probíhala v prostředí programu QGIS, kdy byla jednotlivým vrstvám nastavena vhodná symbologie a poté byly vrstvy pomocí QTiles pluginu vyexportovány jako XYZ dlaždice ve formátu PNG pro vizualizaci na webu. Pro vizualizaci na webu byla využita JavaScriptová knihovna pro tvorbu webových map Leaflet a pro rozdělení na dvě mapová okna a jejich simultánní ovládání byla využita knihovna D3. Uživatel této webové aplikace tak může snadno porovnávat využití taxi služby v jednotlivých letech v měsíci lednu. Pomocí takto vizualizovaných dat, lze pozorovat úbytek žlutých taxi (patrné od roku 2013), který souvisí se zavedením zelených Boro taxi a rozmachem FHV, mezi které patří služby jako Uber, Lyft, Juno, Via apod. Webová aplikace je dostupná na webových stránkách vytvořených v rámci bakalářská práce.



Obr. 16 Vstupní data v programu QGIS a výsledná vizualizace agregovaných dat

V rámci práce byla otestována i rychlost zpracování agregace bodů na dvou sestavách. První sestava (D2S_V3) o konfiguraci 1 jádro, 2 vlákna (Intel® Xeon® Processor E5-2673 v4 @ 2.30 GHz), 8 GB RAM, 4000 IOPS (Input/output operations per second). Druhá testovaná sestava (DS4_V2) byla tvořena konfigurací 8 jader, 8 vláken (Intel® Xeon® Processor E5-2697 v4 @ 2.30 GHz), 28 GB RAM, 32000 IOPS.

První testovací sadou byla data poskytnutá NYC TLC z roku 2010 za měsíc leden. Velikost vstupního souboru činila přibližně 2,5 GB (15 miliónů záznamu). Druhou testovací sadou byla uměle vytvořená datová sada, která vznikla spojením čtyř měsíců – leden, únor, březen a duben. Celková velikost tohoto CSV souboru činila přibližně 10 GB (65 miliónů záznamů). Testovacím kritériem bylo rozlišení výsledné čtvercové sítě (jednotlivých buněk) s agregovanými daty. Konkrétně se jednalo o rozlišení 0.01, 0.005, 0.001, 0.0005 a 0.0001 desetinných stupňů. Testován byl vliv použité konfigurace ve spojení s nastaveným nižším či vyšším rozlišením. Testování bylo provedeno celkem pětkrát pro každé rozlišení. Naměřené výsledky byly zpracovány ve formě tabulek (Příloha 1) a vypočítán byl také aritmetický průměr a medián.



Obr. 17 Srovnání časové náročnosti agregace

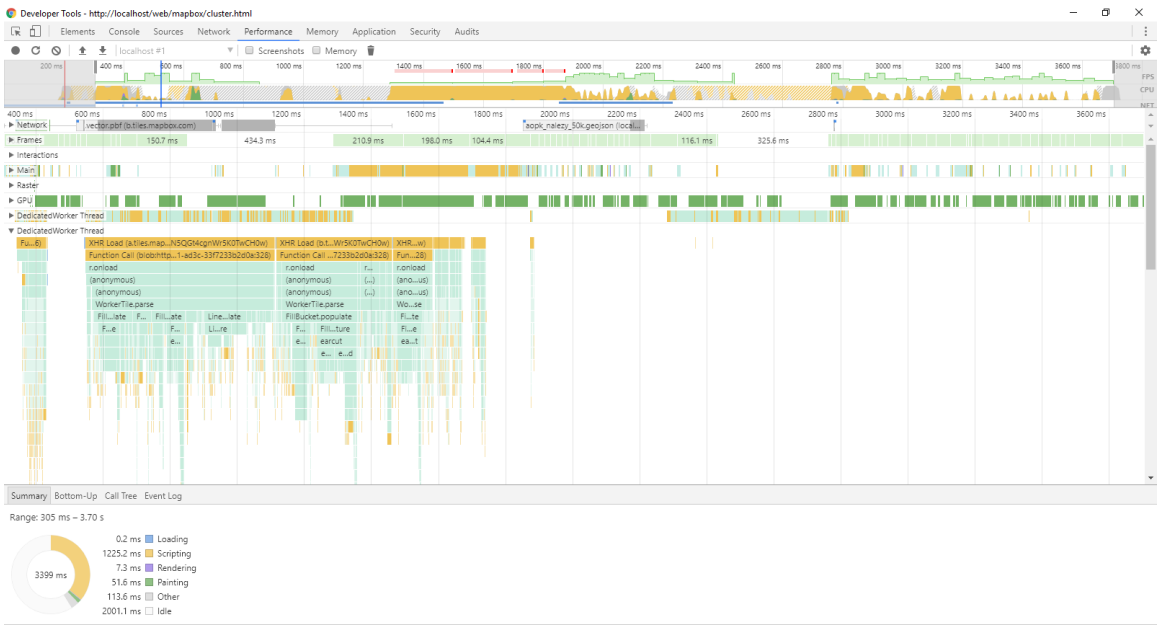
Při testování rychlosti zpracování CSV souboru o velikosti 2,5 GB (přibližně 15 mil. záznamů) na sestavě DS2_V3 za použití nejnižšího testovaného rozlišení (0.01 desetinných stupňů) byla agregace dokončena v průměru za 105,927 sekund. Nejdéle trvala agregace v průměru 174,331 sekund, a to při nastavení nejvyššího testovaného

rozlišení (0.0001 desetinných stupňů). Rozdíl mezi rozlišením 0.005 a 0.001 desetinných stupňů činil průměrně 5,627 sekund. Mezi rozlišením 0.001 a 0.0005 desetinných stupňů se rozdíl pohyboval průměrně kolem 10,756 sekund. Zpracování CSV souboru o velikosti přibližně 10 GB (65 mil. záznamů) na sestavě DS2_V3 trvalo při použití nejnižšího rozlišení v průměru 371,353 sekund. Rozdíl mezi rozlišením 0.005 a 0.001 činil 19,87 sekund a mezi rozlišením 0.001 a 0.0005 vzrostl na 104,426 sekund. Při použití nejvyššího rozlišení byla agregace dokončena průměrně za 561,329 sekund.

Použití výkonnější konfigurace DS4_V2 přineslo značné snížení výpočetní doby. CSV soubor o velikosti 2,5 GB byl na této konfiguraci zpracován za 39,879 sekund při použití nejnižšího rozlišení a při použití nejvyššího rozlišení za 70,342 sekund. Časový rozdíl mezi rozlišením 0.005 a 0.001 byl zanedbatelný (1,313 sekund) a mezi rozlišením 0.001 a 0.0005 rozdíl vzrostl pouze nepatrně na 2,514 sekund. Při testování na CSV souboru o velikosti 10 GB byla agregace při nejnižším rozlišení dokončena průměrně za 137,807 sekund. S použitím nejvyššího rozlišení vzrostl výsledný čas agregace na 250,791 sekund. Rozdíl mezi rozlišením 0.005 a 0.001 činil pouze 6,214 sekund. Mezi rozlišením 0.001 a 0.0005 pak vzrostl na 71,788 sekund.

8 TESTOVÁNÍ JAVASCRIPTOVÝCH KNIHOVEN

V této části bakalářské práce bylo otestováno celkem 5 JavaScriptových knihoven pro vizualizaci dat na webu. V případě shlukování byly otestovány knihovny Leaflet.markercluster, OpenLayers, Supercluster, MapBox GL JS a PruneCluster. Pro tvorbu heatmap bylo otestováno možností knihoven Leaflet a OpenLayers. Porovnávání jednotlivých JavaScriptových knihoven probíhalo na běžné sestavě o konfiguraci Intel® Core™ i5-6200U (2.30 GHz), 8 GB RAM. Jako prohlížeč byl pro testování zvolen Google Chrome ve verzi 65.0.3325.181 a pro měření doby vykreslení mapy bylo využito integrovaných nástrojů pro vývojáře v prohlížeči Google Chrome (karta Performance). Tyto nástroje umožňují přesně sledovat průběh zpracování stránky od prvotního načtení přes skriptování až po její celkové vykreslení (Obr. 18). Testování bylo provedeno na lokálním webovém serveru Apache HTTP Server 2.4.29 (součást balíku XAMPP 7.2.3). Pro analyzování rychlosti načítání webových stránek existují i další specializované nástroje, mezi které můžeme zařadit například Apache Bench či online nástroj webpagetest.org.



Obr. 18 Nástroje pro vývojáře v prohlížeči Google Chrome.

Pro účely testování bylo vytvořeno celkem 9 vzorků dat o různých velikostech (viz Tab. 5), ze kterých byly odebrány nepotřebné atributy. Základem těchto datových sad byl export z Nálezové databáze ochrany přírody od AOPK ČR. Konkrétně se jednalo o vzorky ve formátu GeoJSON o velikostech 10 000, 25 000, 50 000, 100 000, 250 000, 500 000, 1 000 000, 1 500 000 bodů a celý export, který obsahoval 3 127 866 bodů (viz Tab. 1). Jelikož knihovna PruneCluster formát GeoJSON nepodporuje, byla tato knihovna otestována s daty ve formátu JSON. Testována byla rychlost vykreslení celé webové mapy, a to celkem desetkrát pro každý vzorek dat. Z naměřených hodnot byl poté vypočítán aritmetický průměr a medián. Výsledky byly poté zpracovány ve formě tabulek (veškeré naměřené hodnoty jsou uváděny v milisekundách).

Tab. 1 Velikosti testovaných souborů ve formátech JSON a GeoJSON

Formát	Velikost testovaného souboru								
	10 tis.	**25 tis.**	**50 tis.**	**100 tis.**	**250 tis.**	**500 tis.**	**1 mil.**	**1,5 mil.**	**~3 mil.**
JSON	240 kB	600 kB	1,2 MB	2,4 MB	6 MB	12 MB	24 MB	36 MB	75,1 MB
GeoJSON	1,2 MB	2,9 MB	5,8 MB	11,6 MB	28,9 MB	57,9 MB	115,8 MB	173,7 MB	362,1 MB

8.1 Marker Clustering

Leaflet.markercluster

První testovanou JavaScriptovou knihovnou byl plugin Leaflet.markercluster pro Leaflet. Leaflet je jednou z nejznámějších open-source (FreeBSD licence) JavaScriptových knihoven pro tvorbu interaktivních webových map, jejímž autorem je Vladimir Agafonkin. První verze 0.1 vyšla v roce 2011 a aktuálně je Leaflet dostupný ve verzi 1.3.1. Je navržen tak, aby obsahoval pouze základní funkcionalitu (na rozdíl od OpenLayers) a byl dále rozšiřován pomocí zásuvných modulů (pluginů). Leaflet funguje na všech hlavních desktopových a mobilních platformách a pro svou funkcionalitu využívá technologii HTML 5 a CSS3. Zásuvný modul Leaflet.marketcluster vyšel v první verzi 0.1 v roce 2012 (aktuální verze 1.3.0 z ledna 2018) a jeho autorem je Dave Leaver (Leaflet, 2018).

Podle naměřených hodnot je řešení, které poskytuje plugin Leaflet.markercuster nejpomalejší v celém testování. Při testování na vzorku o velikosti 10 tisíc bodů bylo zjištěno, že řešení dosahuje srovnatelných časů (1572 ms) s knihovnou OpenLayers (1388 ms). Od 50 tisíc bodů již ale rapidně narůstá doba vykreslení (11612,5 ms) a lze pozorovat zvýšenou latenci při překreslování mezi jednotlivými úrovněmi přiblížení, především pak při přechodu z větších úrovní přiblížení do nižších. Možnou příčinou je použití animace překreslování na úkor rychlosti. Limitem pro tuto knihovnu byla datová sada o velikosti 100 tisíc bodů, která byla v průměru vykreslena za 47154,3 ms. Pro objemnější datové sady nebylo možné testování dokončit z důvodu „zamrzání“ a padání webového prohlížeče.

Tab. 2 Výsledky testování knihovny Leaflet.markercluster (naměřené hodnoty jsou uvedeny v milisekundách)

Měření	Počet bodů v datové sadě								
	10 tis.	**25 tis.**	**50 tis.**	**100 tis.**	**250 tis.**	**500 tis.**	**1 mil.**	**1,5 mil.**	**~3 mil.**
1.	1549	3439	12701	47252	N/A	N/A	N/A	N/A	N/A
2.	1677	3373	11167	46872	N/A	N/A	N/A	N/A	N/A
3.	1576	3371	11056	46966	N/A	N/A	N/A	N/A	N/A
4.	1564	3501	11093	46914	N/A	N/A	N/A	N/A	N/A
5.	1590	3326	11176	48109	N/A	N/A	N/A	N/A	N/A
6.	1582	3399	12419	47442	N/A	N/A	N/A	N/A	N/A
7.	1531	3457	11810	46161	N/A	N/A	N/A	N/A	N/A
8.	1560	3467	11093	47323	N/A	N/A	N/A	N/A	N/A
9.	1521	3367	12380	46666	N/A	N/A	N/A	N/A	N/A
10.	1567	3344	11230	47838	N/A	N/A	N/A	N/A	N/A
Arit. průměr	1571,7	3404,4	11612,5	47154,3	N/A	N/A	N/A	N/A	N/A
Medián	1565,5	3386	11203	47109	N/A	N/A	N/A	N/A	N/A

OpenLayers

Druhou testovanou JavaScriptovou knihovnou je OpenLayers, která je taktéž licencována pod FreeBSD licencí. První verze byla vyvinuta společností MetaCarta a vyšla již v roce 2006 (OpenLayers, 2018). OpenLayers nativně podporuje shlukování bodů, takže není nutné využívat jiné zásuvné moduly. Ty nicméně existují a rozšiřují nativní shlukování v OpenLayers například o animované přechody mezi jednotlivými úrovněmi přiblížení obdobně jako u Leafletu (plugin ol-ext, původně samostatný plugin OL3-AnimatedCluster).

Shlukování v knihovně OpenLayers dosahuje relativně dobrých výsledků. Při testování na datové sadě o velikosti 250 tisíc bodů, byl průměrný čas vykreslení 8061 ms. Při testování na datové sadě o velikosti 500 tisíc bodů došlo k navýšení času vykreslení na trojnásobek (průměrně 24455 ms). Objemnější datové sady již nedokázala knihovna OpenLayers vykreslit a veškeré pokusy končily pádem či „zamrznutím" prohlížeče.

Tab. 3 Výsledky testování knihovny OpenLayers (naměřené hodnoty jsou uvedeny v milisekundách)

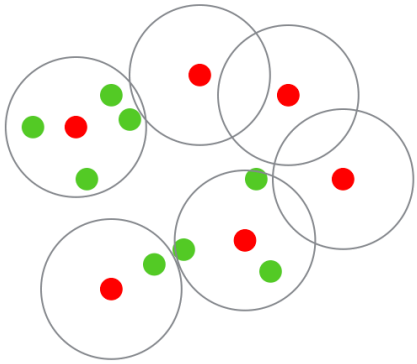
Měření	**Počet bodů v datové sadě**								
	10 tis.	**25 tis.**	**50 tis.**	**100 tis.**	**250 tis.**	**500 tis.**	**1 mil.**	**1,5 mil.**	**~3 mil.**
1.	1447	1810	2521	4616	8283	24926	N/A	N/A	N/A
2.	1458	2046	2749	4543	8149	24160	N/A	N/A	N/A
3.	1262	1941	2507	4532	8303	24392	N/A	N/A	N/A
4.	1419	1857	2561	4696	8147	25013	N/A	N/A	N/A
5.	1299	1948	2738	4326	8024	23883	N/A	N/A	N/A
6.	1360	2045	2757	4526	8241	24127	N/A	N/A	N/A
7.	1435	1961	2734	4381	8000	24115	N/A	N/A	N/A
8.	1514	1914	2564	4191	7953	24586	N/A	N/A	N/A
9.	1416	2081	2615	4362	7919	24463	N/A	N/A	N/A
10.	1266	1942	2415	4333	8061	24885	N/A	N/A	N/A
Arit. průměr	1387,6	1954,5	2616,1	4450,6	8108	24455	N/A	N/A	N/A
Medián	1417,5	1945	2589,5	4453,5	8104	24427,5	N/A	N/A	N/A

Supercluster

Další testovanou knihovnou je knihovna Supercluster. Autorem této knihovny je Vladimir Agafonkin (autor knihovny Leaflet). Supercluster je samostatná knihovna, která může být použita v kombinaci s jinou libovolnou JavaScriptovou knihovnou pro tvorbu webových map. V tomto případě byla otestována její rychlost ve spojení s knihovnou Leaflet. Tato knihovna pro tvorbu shluků využívá metodu *hierarchical greedy clustering*. Tvorba shluku začíná vybráním bodu z datasetu, kolem kterého jsou nalezeny všechny body v rámci zvoleného poloměru, ze kterých je vytvořen shluk. Tvorba dalšího shluku začíná vybráním bodu, který ještě součástí žádného shluku není. Tuto metodu využívá i již dříve zmíněný Leaflet.markercluster plugin pro Leaflet. V knihovně Supercluster je ale tato metoda navíc rozšířena o tzv. prostorový index, kdy jsou body jen jednou zpracovány do speciální datové struktury, kterou lze poté okamžitě využít pro pozdější dotazování (Agafonkin, 2016).

Vzorek o velikosti 1,5 miliónů bodů vykreslila knihovna Supercluster za 23911,5 ms (řešení OpenLayers bylo schopné vykreslit, v přibližně stejném čase 24455 ms, pouze export o poloviční velikosti). Tato knihovna byla schopna bez problému vykreslit

webovou mapu i při použití celého exportu z ND OP. Celý export byl vykreslen za 71776,5 ms. Přechody mezi jednotlivými úrovněmi přiblížení byly rychlé, nicméně nedosahovaly plynulosti knihovny Mapbox GL JS.



Obr. 19 Princip metody Hierarchical Greedy Clustering (zdroj: https://blog.mapbox.com/).

Tab. 4 Výsledky testování knihovny Supercluster (naměřené hodnoty jsou uvedeny v milisekundách)

Měření	**Počet bodů v datové sadě**								
	10 tis.	**25 tis.**	**50 tis.**	**100 tis.**	**250 tis.**	**500 tis.**	**1 mil.**	**1,5 mil.**	**~3 mil.**
1.	958	1150	1514	2483	5401	12461	23680	35957	72563
2.	952	1152	1499	2527	5283	12554	23632	36255	72097
3.	991	1153	1464	2497	5389	12207	24165	36685	71360
4.	768	1103	1587	2533	5271	12365	23670	36377	71120
5.	853	1179	1538	2537	5327	12239	23707	35103	72442
6.	828	1189	1475	2525	5242	12555	24632	36333	71465
7.	810	1166	1492	2480	5228	12525	23821	36203	72291
8.	782	1129	1484	2568	5281	12402	24418	35639	71576
9.	995	1096	1552	2597	5331	12443	22971	35750	71238
10.	819	1165	1446	2503	5253	12459	24419	36339	71613
Arit. průměr	875,6	1148,2	1505,1	2525	5300,6	12421	23911,5	36064,1	71776,5
Medián	840,5	1152,5	1495,5	2526	5282	12451	23764	36229	71594,5

Mapbox GL JS

Předposlední knihovnou, ve které byla otestována funkcionalita shlukování bodů je knihovna Mapbox GL JS. Tato knihovna využívá pro shlukování již zmíněnou knihovnu Supercluster, nicméně Mapbox GL JS používá navíc pro vykreslování technologii WebGL, díky které je využito pro vykreslení GPU. Vzhledem k tomu, že pro shluování bodů je použita knihovna Supercluster by se dalo očekávat, že knihovna Mapbox GL JS bude dosahovat obdobných výsledků jako knihovna Supercluster v kombinaci s knihovnou Leaflet.

Do velikosti 100 tisíc bodů dosahuje toto řešení horších výsledků než knihovna Supercluster v kombinaci s knihovnou Leaflet. S použitím objemnějších datových sad se ale rozdíl ztrácí a při testování se vzorkem o velikosti 250 tisíc nálezu byla již rychlost vykreslení srovnatelná (rozdíl pouze 167,8 ms), Při použití vzorku o velikosti 500 tisíc nálezu byla knihovna Mapbox GL JS již o 4329,7 ms rychlejší. Celý export z ND OP pak byla schopna vykreslit průměrně za 34224,8 ms (oproti 71775,6 ms, kterých dosahovalo řešení Supercluster a Leaflet). Tato knihovna byla ze všech testovaných nejplynulejší a překreslení při změnách úrovní přiblížení bylo okamžité, a to i při použití celého exportu ND OP.

Tab. 5 Výsledky testování knihovny Mapbox GL JS (naměřené hodnoty jsou uvedeny v milisekundách)

Měření	Počet bodů v datové sadě								
	10 tis.	**25 tis.**	**50 tis.**	**100 tis.**	**250 tis.**	**500 tis.**	**1 mil.**	**1,5 mil.**	**~3 mil.**
1.	3083	3378	3393	4035	5505	8019	13007	17665	33964
2.	2850	3267	3390	3942	5484	8180	12906	17889	34378
3.	2786	3120	3302	4106	5397	8140	12958	17404	33835
4.	2824	3332	3301	4073	5474	8175	12768	17678	34097
5.	3087	3055	3450	4114	5475	8157	13457	17704	34677
6.	3029	3194	3473	4078	5462	8045	13011	17811	34459
7.	2875	3149	3501	4017	5520	8025	13121	17460	34719
8.	2970	3292	3345	4010	5397	8152	13029	17486	33449
9.	2895	3123	3411	4041	5464	8059	13010	17220	34469
10.	3036	3345	3402	3973	5506	7961	13410	17668	34201
Arit. průměr	2943,5	3225,5	3396,8	4038,9	5468,4	8091,3	13067,7	17598,5	34224,8
Medián	2932,5	3230,5	3397,5	4038	5474,5	8099,5	13011,5	17665,5	34289,5

PruneCluster

PruneCluster je posledním testovaným řešením (plugin pro knihovnu Leaflet) a jedná se o projekt norské organizace SINTEF (Stiftelsen for industriell og teknisk forskning). Autory této knihovny jsou Antoine Pultier a Aslak Wegner Eide. Tato knihovna poskytuje efektivní řešení shlukování bodů ve webových mapových aplikacích. Autoři této knihovny vyvinuli nový algoritmus pro shlukování a aktualizaci bodů v reálném čase. Inspirací jim byly algoritmy pro detekci kolizí dvou objektů. Použitý algoritmus, podle testování autorů, vykazuje značné zlepšení oproti jiným dostupným řešením. Knihovna je tak vhodná pro vizualizaci velkých datasetů a to dokonce i v reálném čase. Mezi výhody patří také možnost členit jednotlivé body do kategorií (Obr. 20) a zobrazit pak jejich zastoupení ve shluku (SINTEF, 2018). Knihovna PruneCluster nepodporuje formát GeoJSON a proto byla testována s daty ve formátu JSON. Data v tomto formátu mají odlišnou strukturu a podstatně nižší velikost (72,1 MB oproti 362,1 MB v případě celého exportu z ND OP). Proto nelze knihovnu PruneCluster přímo srovnávat s ostatními testovanými knihovnami. Pro shlukování ale používá odlišný algoritmus, a tak byla do testování zařazena.

Knihovna v testování dosahovala nejnižších časů, ze všech testovaných knihoven, kdy půl miliónu bodů zvládla vykreslit průměrně za 3202,9 ms. Celý export z ND OP byl pak v průměru vykreslen za 23111,9 ms. Z výsledků testu lze usuzovat, že v případě použití GeoJSON formátu, by pravděpodobně poskytovala nejrychlejší čas vykreslení knihovna Mapbox GL JS (23111,9 ms oproti 34224,8 ms). Knihovna PruneCluster, obdobně jako Leaflet.markercluster, využívá animaci při přechodu mezi jednotlivými úrovněmi přiblížení. Při použití menších vzorků dat je odezva při překreslení únosná, nicméně při použití větších vzorků (250 tisíc bodů a více) se použitá animace již negativně projevuje na rychlosti.

Obr. 20 Zastoupení kategorií ve shluku (zdroj: https://github.com/SINTEF-9012/PruneCluster).

Tab. 6 Výsledky testování knihovny PruneCluster (naměřené hodnoty jsou uvedeny v milisekundách)

Měření	**Počet bodů v datové sadě**								
	10 tis.	**25 tis.**	**50 tis.**	**100 tis.**	**250 tis.**	**500 tis.**	**1 mil.**	**1,5 mil.**	**~3 mil.**
1	794	879	945	1169	1891	3319	7668	14923	23425
2	769	817	920	1198	1864	3213	7284	13759	22273
3	761	827	920	1192	1952	3175	7553	14746	23107
4	764	847	936	1265	1880	3096	7615	14929	22625
5	761	804	935	1136	1967	3153	7352	13858	22850
6	763	809	931	1196	1836	3214	7307	14480	22753
7	747	850	978	1260	1931	3187	7658	14443	22978
8	765	856	932	1153	1867	3240	7736	13497	23747
9	775	804	973	1116	1904	3202	7541	14596	23872
10	769	835	935	1174	1934	3230	7732	14506	23489
Arit. průměr	766,8	832,8	940,5	1185,9	1902,6	3202,9	7544,6	14373,7	23111,9
Medián	764,5	831	935	1183	1897,5	3207,5	7584	14493	23042,5

8.2 Heatmap

Testováním JavaScriptových knihoven pro tvorbu heatmap se zabýval Ježek a kol. (2017). Ve svém příspěvku porovnává rychlost vykreslování heatmapy pomocí Google Maps JavaScript API, Leaflet.heat pluginu pro Leaflet, ArcGIS online a vlastní WebGL řešení – WebGLayer, které vzniklo na Západočeské univerzitě v Plzni. Poslední zmíněné řešení dosahuje mnohem nižších vykreslovacích časů než řešení předešlá (100 ms pro datovou sadu o velikosti 1 492 475 bodů).

Pro vykreslování heatmap byly v rámci testovány dvě knihovny. První je plugin Leaflet.heat ve verzi 0.2.0 pro knihovnu Leaflet, který využívá knihovny simpleheat.js (inspirací pro tuto knihovnu byl plugin heatmap.js). Druhou testovanou knihovnou je OpenLayers, která tvorbu heatmapy podporuje nativně. U obou testovaných knihoven probíhá vykreslení heatmapy na HTML canvas. U heatmap lze nastavit parametry jako rádius, ve kterém má docházet ke sčítání vlivu bodů či rozostření. Podle poznatků Ježka a kol. (2017) nemají tyto parametry vliv na výsledný čas vykreslení heatmapy.

Leaflet.heat

Knihovna Leaflet v kombinaci s pluginem pro tvorbu heatmap Leaflet.heat byla schopna vykreslit všechny testovací vzorky. V porovnání s knihovnou OpenLayers bylo řešení Leaflet.heat schopno vykreslit desetinásobně více bodů ve srovnatelném čase (průměrně 1217,8 ms pro 100 tisíc bodů při použití Leaflet.heat oproti 1276 ms pro 10 tisíc bodů při použití OpenLayers). Celý export z ND OP tedy 3 127 866 bodů bylo toto řešení schopno vykreslit v průměru za 16313,7 ms.

Tab. 7 Výsledky testování knihovny Leaflet.heat (naměřené hodnoty jsou uvedeny v milisekundách)

Měření	Počet bodů v datové sadě								
	10 tis.	**25 tis.**	**50 tis.**	**100 tis.**	**250 tis.**	**500 tis.**	**1 mil.**	**1,5 mil.**	**3 127 866**
1.	825	900	1104	1350	1821	3432	5417	8672	18456
2.	856	902	1049	1252	1819	3206	5631	8861	16260
3.	946	892	1037	1194	1832	3330	5555	8146	15937
4.	901	925	1153	1191	1822	3032	5524	8052	15614
5.	808	952	1032	1143	1794	3216	5550	8740	15865
6.	793	887	1068	1189	1860	3088	5754	8501	16136
7.	885	957	1004	1190	1837	3171	5947	8062	16225
8.	867	875	1012	1221	1795	3109	5682	7966	16432
9.	986	793	985	1225	1842	3121	5607	8578	16304
10.	746	952	1052	1223	1775	3256	5545	8241	15908
Arit. průměr	861,3	903,5	1049,6	1217,8	1819,7	3196,1	5621,2	8381,9	16313,7
Medián	861,5	901	1043	1207,5	1821,5	3188,5	5581	8371	16180,5

OpenLayers

Knihovna OpenLayers byla schopna vykreslit maximálně 250 tisíc bodů, a to průměrně za 10846,5 ms. S objemnějšími vzorky dat nebyla knihovna schopna pracovat a veškeré další testy vyústily v „zamrznutí“ či pád webového prohlížeče. I při použití nejmenšího testovacího vzorku vykazovala webová aplikace při uživatelské interakci zvýšenou odezvu a manipulace s mapou nebyla plynulá.

Tab. 2 Výsledky testování knihovny OpenLayers (naměřené hodnoty jsou uvedeny v milisekundách)

Měření	Počet bodů v datové sadě								
	10 tis.	**25 tis.**	**50 tis.**	**100 tis.**	**250 tis.**	**500 tis.**	**1 mil.**	**1,5 mil.**	**3 127 866**
1.	1256	1977	3289	4763	11358	N/A	N/A	N/A	N/A
2.	1413	1759	2995	4686	11258	N/A	N/A	N/A	N/A
3.	1143	1811	2910	5109	10025	N/A	N/A	N/A	N/A
4.	1266	1861	2888	4931	11002	N/A	N/A	N/A	N/A
5.	1438	1982	3152	4936	10416	N/A	N/A	N/A	N/A
6.	1328	1753	3155	4825	11029	N/A	N/A	N/A	N/A
7.	1474	1826	2939	5184	10146	N/A	N/A	N/A	N/A
8.	1160	1686	2883	5038	11154	N/A	N/A	N/A	N/A
9.	1154	1845	3017	5072	10889	N/A	N/A	N/A	N/A
10.	1128	1878	2889	4925	11188	N/A	N/A	N/A	N/A
Arit. průměr	1276	1837,8	3011,7	4946,9	10846,5	N/A	N/A	N/A	N/A
Medián	1261	1835,5	2967	4933,5	11015,5	N/A	N/A	N/A	N/A

8.3 Vektorové dlaždice

Další možností vizualizace dat je zobrazení celého datasetu za využití vektorových dlaždic. Díky tomuto přístupu lze efektivně zobrazit celé objemné datové sady včetně zachování interaktivity (lze přistupovat k jednotlivým atributům). Vektorové dlaždice lze vytvořit z dat ve formátu JSON, GeoJSON, CSV či Geobuf nástrojem tippecanoe, který je ale dostupný pouze pro operační systémy Linux či macOS. Na operačním systému macOS lze pro nainstalovaní nástroje tippecanoe využít správce softwarových balíčku

homebrew. Instalace se poté provede příkazem `brew install tippecanoe` v terminálu. V prostředí operačního systému Linux je nutné nástroj tippecanoe ručně zkompilovat ze zdrojového repozitáře.

Po nainstalování lze vytvořit vektorové dlaždice příkazem `tippecanoe –o vystupni_soubor.mbtiles [paramtery] vstupni_soubor.geojson`. Výstupem bude soubor MBTiles, který obsahuje výsledné vektorové dlaždice (po technické stránce se jedná o SQLite databázi pro ukládání vektorových či rastrových dlaždic). Vektorové dlaždice je možné také vyexportovat samostatně pro jednotlivé úrovně přiblížení, a to použitím parametru `–e vystupni_adresar` namísto parametru `–o`. Výsledkem pak budou jednotlivé vektorové dlaždice ve formátu PBF (Protocolbuffer Binary Format) Nastavením parametrů lze ovlivnit maximální či minimální úrovně přiblížení, pro které se mají dlaždice generovat, míru generalizace apod. Pro vizualizaci je poté možné vektorové dlaždice nahrát na uložiště poskytované společností Mapbox a vizualizovat pomocí JavaScriptové knihovny Mapbox GL JS. Další možností, jak lze výsledné vektorové dlaždice hostovat je využití nástroje TileServer pro spuštění vlastního tileserveru.

9 VÝSLEDKY

Cílem práce bylo provést analýzu paradigmata Big Data a na zvolených případech demonstrovat odlišnosti oproti konvenčnímu zpracování geodat. Pro tyto účely bylo pracováno s Apache Hadoop distribucí HDP Sandbox 2.5 na virtuálním počítači platformy Microsoft Azure. Byla ověřena funkcionalita zvoleného balíku nástrojů pro zpracování prostorových Big Data, kterým byl balík GIS Tools for Hadoop. Funkcionalita toho balíku nástrojů byla otestována na dvou příkladech. Prvním příkladem pro otestování základní funkcionality byla agregace bodů v polygonu, jejíž výsledkem bylo početní zastoupení zemětřesení v dílčích okresech Kalifornie. Druhým příkladem byla agregace CSV dat, konkrétně taxi záznamů z New York City, do čtvercové sítě. Celý postup práce je popsán v kapitole 7. Testován byl vliv rozlišení čtvercové sítě na rychlost zpracování agregace. Pro účely testování byla použita dvojice CSV souborů s taxi daty o velikosti přibližně 2,5 GB (15 miliónů záznamů) a přibližně 10 GB (65 milionů záznamů). Testovaní probíhalo na dvou sestavách o odlišných konfiguracích. První sestava byla tvořena konfigurací 1 jádro, 2 vlákna (Intel® Xeon® Processor E5-2673 v4 @ 2.30 GHz), 8 GB RAM, 4000 IOPS a druhá 8 jader, 8 vláken (Intel® Xeon® Processor E5-2697 v4 @ 2.30 GHz), 28 GB RAM, 32000 IOPS. Agregace do čtvercové sítě byla provedena celkem pětkrát pro každé testované rozlišení.

Při použití CSV souboru o velikosti 2,5 GB byla agregace dokončena na první testované sestavě za 105,927 sekund při použití nejnižšího rozlišení (0.01 desetinných stupňů) a za 174,331 sekund při nastavení nejvyššího testovaného rozlišení (0.0001 desetinných stupňů). Zpracování CSV souboru o velikosti 10 GB trvalo na první testované sestavě 371,353 sekund při nejnižším rozlišení čtvercové sítě a 561,329 sekund při použití nejvyššího rozlišení.

Testování na výkonnější sestavě přineslo značné snížení doby, za kterou byla agregace do čtvercové sítě dokončena. Agregace CSV dat o velikosti 2,5 GB byla dokončena za 39,879 sekund při použití nejnižšího rozlišení a za 70,342 sekund v případě použití nejvyššího rozlišení. Zpracování CSV dat o velikosti 10 GB na této sestavě poté proběhlo za 137,807 sekund při nejnižším rozlišení. Použitím nejvyššího rozlišení vzrostl potřebný čas pro zpracování na 250,791 sekund.

Data ze žlutých taxi (leden 2009–2016) a zelených Boro taxi (leden 2014–2016) byla agregována do čtvercové sítě o rozlišení 0.001 desetinných stupňů, vizualizována pomocí programu QGIS a následně vyexportována jako XYZ dlaždice. Výsledná mapová aplikace, která umožňuje srovnání využití taxi v letech 2009–2016 v měsíci lednu je součástí webových stránek, které byly vytvořeny v rámci bakalářské práce.

Dále byly v praktické části práce otestovány JavaScriptové knihovny pro shlukování bodů a tvorbu heatmap. Testována byla jejich schopnost vizualizovat objemné datové sady. Testování probíhalo na běžné sestavě o konfiguraci Intel® Core™ i5-6200U (2.30 GHz), 8 GB RAM a lokálním webovém serveru Apache HTTP Server 2.4.29. Jako prohlížeč byl pro testování zvolen Google Chrome ve verzi 65.0.3325.181. Vzorky dat pro testování byly vytvořeny z exportu ND OP.

Shlukování bodů bylo otestováno na pěti JavaScriptových knihovnách. Knihovna Leaflet.markercluster byla schopna vykreslit nanejvýše vzorek o velikosti 100 tisíc bodů, a to v čase 47154,3 ms. Objemnější vzorky již nebyla knihovna schopna vykreslit. Jednalo se tak o nejpomalejší knihovnu v tomto testování. Knihovna OpenLayers podporuje shlukování nativně a byla schopna vykreslit maximálně vzorek o velikosti 500 tisíc bodů. Tento vzorek byl vykreslen v čase 24455 ms. Testování na dalších

vzorcích nebylo možné dokončit z důvodu padání webového prohlížeče. Knihovna Supercluster v kombinaci s knihovnou Leaflet již byla schopna vykreslit celý export z ND OP (3 127 866 bodů) a to v čase 71776,5 ms. Překreslování při změně úrovní bylo rychlé, nicméně nedosahovalo takové plynulosti jako v případě další testované knihovny Mapbox GL JS. Tato knihovna byla schopna vykreslit celý export během 34224,8 ms. Použitím technologie WebGL, která pro vykreslení využívá GPU, byla interakce s webovou mapou nejplynulejší ze všech testovaných knihoven. Poslední testovanou knihovnou byla knihovna PruneCluster. Tato knihovna nepodporuje formát GeoJSON, a proto byla otestována se vzorky ve formátu JSON. Celý export dokázala knihovna PruneCluster vykreslit za 23111,9 ms. V případě použití dat ve formátu GeoJSON by doba nutná pro vykreslení pravděpodobně výrazně vzrostla, a to z důvodu, že data ve formátu GeoJSON jsou objemnější (362,1 MB) než ve formátu JSON (75,1 MB).

Pro tvorbu heatmap byla otestována knihovna Leaflet.heat a knihovna OpenLayers, která stejně jako shlukování bodů, podporuje nativně i tvorbu heatmap. První zmíněná knihovna vykreslila heatmapu, při použití celého exportu, během 16313,7 ms. Knihovna OpenLayers nebyla schopna celý export vykreslit. Limitem pro tuto knihovnu byl vzorek o velikosti 250 tisíc bodů, který byl vykreslen za 10846,5 ms. I při použití nejmenšího vzorku vykazovala knihovna zvýšenou odezvu, která se s použitím objemnějších vzorků zvyšovala.

Vytvořené webové aplikace pro shlukování bodů a tvorbu heatmap jsou dostupné na webových stránkách vytvořených pro potřeby této bakalářské práce.

10 DISKUZE

Cílem této bakalářské práce bylo teoreticky specifikovat a ověřit možnosti vizualizace vybraných zdrojů dat, tedy konkrétní metody, technologie a postupy. Balík nástrojů GIS Tools for Hadoop byl vybrán, jelikož umožňuje propojení Apache Hadoop a programu ArcMap z ArcGIS Desktop pomocí Geoprocessing Tools for Hadoop toolboxu. Tento balík byl testován v prostředí HDP Sandbox na virtuálním počítači platformy Microsoft Azure.

První problém se týkal Geoprocessing Tools for Hadoop toolboxu, konkrétně nástroje *Copy from HDFS*. Jelikož byl HDP Sandbox hostován na vzdáleném serveru platformy Microsoft Azure a nástroj *Copy from HDFS* neumožňuje připojení přes proxy, nebylo možné tento nástroj využít pro export agregovaných dat z HDFS. Alternativním řešením bylo využití správce souborů pro prohlížení HDFS, který je integrován do webového rozhraní pro správu Ambari.

Dalším problémem byly chybějící buňky (patrné při přiblížení na oblast Manhattanu) v jedné z výsledných vrstev, konkrétně *yellow_tripdata_2012_01_dropoffs_merged* (agregovaná výstupní místa žlutých taxi v lednu 2012). Jedná se pravděpodobně o chybu ve vstupních datech, jelikož i při opětovném provedení agregace do čtvercové sítě tyto buňky chyběly.

Ke clusterovému řešení nebylo přistoupeno, jelikož pro účely práce, tedy demonstraci odlišností paradigmata Big Data oproti konvenčnímu zpracování geodat, plně postačovalo prostředí HDP Sandbox. Využití clusteru na konkrétním příkladu a za konkrétním účelem by bylo vhodnou možností, jak navázat na tuto práci. Otázkou je, zdali by bylo využití clusterového řešení výhodné a výsledný čas se oproti single-node řešení spíše neprodloužil (viz Růžička a kol. 2017). Další možností navázání by mohlo být otestování rozšíření MapReduce frameworku SpatialHadoop, které je určeno pro práci s prostorovými daty. Nabízí se také možnost využití jiného frameworku pro zpracování dat. Příkladem může být Apache Spark, jehož možností využívají například nástroje GeoTrellis.

Další část práce tvořilo testování JavaScriptových knihoven na exportu z ND OP. Nejedná se o Big Data v pravém slova smyslu, nicméně export o velikosti přesahující 3 milióny záznamu vyžaduje využití speciálních nástrojů (v tomto případě specializovaných JavaScriptových knihoven), které dovedou takové množství záznamů zpracovat a vykreslit. Řešení Leaflet.markercluster a OpenLayers se jeví jako zcela nevhodné. Knihovna Leaflet.markercluster dokázala vykreslit maximálně vzorek o velikosti 100 tisíc bodů, a to za 47154,3 ms. Řešení OpenLayers vykreslilo při shlukování bodů maximálně 250 tisíc bodu během 24455 ms. Při tvorbě heatmap pak pouze 250 tisíc bodů za 10846,5 ms.

Webová mapa vykreslená pomocí knihovny Mapbox GL JS byla díky použité technologii WebGL, která využívá pro vykreslení GPU, nejplynulejší ze všech testovaných knihoven. Výhody této technologie potvrzuje i testování rychlosti vykreslení heatmap (viz Ježek a kol. 2017), kdy bylo vykreslení při použití technologie WebGL téměř okamžité. Vykreslování takto objemných datových sad lze také urychlit použitím vektorových dlaždic a to i v kombinaci s metodou Marker Clustering.

11 ZÁVĚR

Hlavním cílem této práce bylo provést analýzu paradigmatu Big Data v oblasti GIS/GIT s ohledem na možnosti vizualizace. Samotné tvorbě práce předcházelo studium tohoto tématu, převážně ze zahraničních publikací, na jehož základě byl v teoretické části popsán vznik pojmu Big Data a jeho nejčastěji uváděné charakteristiky, mezi které patří velikost (volume), rychlost (velocity) a různorodost (variety), často označované jako tzv. „3V“.

Dále se autor v práci zaměřil na charakteristiku dílčích komponent Apache Hadoop frameworku, tedy na distribuovaný souborový systém HDFS pro ukládání dat, systém řízení zdrojů YARN a programovací model umožňující paralelní zpracování objemných datových sad MapReduce. Popsány byly i populární distribuce tohoto frameworku, a to konkrétně Cloudera CDH, Hortonworks Data Platform a MapR Converged Data Platform, které reprezentují lokální variantu tohoto frameworku. Implementaci Apache Hadoop v cloudu pak představují distribuce HDInsight či Elastic MapReduce.

V rámci bakalářské práce probíhalo i testování funkcionality vybrané open source sady nástrojů pro práci s prostorovými daty GIS Tools for Hadoop. Testování probíhalo na příkladech agregace bodů v polygonu a agregaci v čtvercové síti, pro kterou byla využita taxi data z New York City. Data ze žlutých taxi (leden 2009–2016) a zelených Boro taxi (leden 2014–2016) byla agregována do čtvercové sítě a vizualizována na webu. Výsledná mapová aplikace byla umístěna na webové stránky, které byly vytvořeny pro účely bakalářské práce.

Poslední část práce tvořilo testování JavaScriptových knihoven pro shlukování bodů a tvorbu heatmap. Shlukování bylo testováno na knihovnách Leaflet.markercluster, OpenLayers, Supercluster, Mapbox GL JS a PruneCluster. Pro tvorbu heatmap byly testovány knihovny Leaflet.heat a OpenLayers. Zmíněno je také proces vytvoření vektorových dlaždic a jejich využití pro vizualizaci na webu.

POUŽITÁ LITERATURA A INFORMAČNÍ ZDROJE

AGAFONKIN, Vladimir, 2016. Clustering millions of points on a map with Supercluster. *Points of interest: The official Mapbox blog* [online]. [cit. 2018-04-09]. Dostupné z: https://blog.mapbox.com/clustering-millions-of-points-on-a-map-with-supercluster-272046ec5c97

Amazon EMR. *Amazon Web Services (AWS) - Cloud Computing Services* [online]. 2018 [cit. 2018-04-11]. Dostupné z: https://aws.amazon.com/emr/

Apache Hadoop YARN. *Welcome to The Apache Software Foundation!* [online]. 2018 [cit. 2018-04-22]. Dostupné z: https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html

Azure HDInsight Documentation. *Microsoft Azure Cloud Computing Platform & Services* [online]. 2018 [cit. 2018-04-11]. Dostupné z: https://docs.microsoft.com/en-us/azure/hdinsight/

BALUSAMY, Balamurugan, Vegesna Tarun Sai VARMA a Sohil Sri Mani Yeshwanth GRANDHI. Challenges in Big Data Analytics. In: SOMANI, Arun K. a Ganesh Chandra DEKA, eds. *Big Data Analytics: Tools and Technology for Effective Planning*. Boca Raton, 2018, s. 38–53. ISBN 978-1-138-03239-2.

BROVELLI, Maria A., Daniele OXOLI a Mayra ZURBARÁN. *Sensing Slow Mobility and Interesting Locations for Lombardy Region (Italy): A Case Study Using Pointwise Geolocated Open Data* [online]. 2016 [cit. 2018-05-02]. Dostupné z: https://www.researchgate.net/publication/303869963_SENSING_SLOW_MOBILITY_AND_INTERESTING_LOCATIONS_FOR_LOMBARDY_REGION_ITALY_A_CASE_STUDY_USING_POINTWISE_GEOLOCATED_OPEN_DATA

CASTRIOTTA, Adriana Grazia. *Sentinel Data Access Annual Report 2016* [online]. 2017 [cit. 2018-03-14]. Dostupné z: https://scihub.copernicus.eu/twiki/pub/SciHubWebPortal/AnnualReport2016/COPE-SERCO-RP-17-0071_-_Sentinel_Data_Access_Annual_Report_2016_v1.2.pdf

Cloudera - Machine Learning | Analytics | Cloud [online]. 2018 [cit. 2018-04-10]. Dostupné z: https://www.cloudera.com/

COX, Michael a David ELLSWORTH. Application-Controlled Demand Paging for Out-of-Core Visualization. In: *Proceedings of the 8th conference on Visualization '97* [online]. [cit. 2018-03-13]. ISBN 1-58113-011-2. Dostupné z: http://www.evl.uic.edu/cavern/rg/20040525_renambot/Viz/parallel_volviz/paging_outofcore_viz97.pdf

CROITORU, Arie, Andrew CROOKS, Jacek RADZIKOWSKI, Anthony STEFANIDIS, Ranga R. VATSAVAI a Nicole WAYANT. Geoinformatics and Social Media: New Big Data Challenge. In: KARIMI, Hassan A., ed. *Big Data: Techniques and Technologies in Geoinformatics*. Boca Raton: CRC Press, Taylor & Francis Group, 2014, s. 207–228. ISBN 978-1-4665-8651-2.

CHEN, Min, Shiwen MAO a Yunhao LIU. Big Data: A Survey. In: *Mobile Networks and Applications* [online]. 2014, **19**(2), s. 171–209 [cit. 2018-04-30]. DOI: 10.1007/s11036-013-0489-0. ISSN 1572-8153. Dostupné z: https://pdfs.semanticscholar.org/a8d5/edc845fe8512e01ddfd4af0d09c397fbcbec.pdf

Data Management Platform, Solutions and Big Data Analysis | Hortonworks [online]. 2018 [cit. 2018-04-30]. Dostupné z: https://hortonworks.com/

DEAN, Jeffrey a Sanjay GHEMAWAT. *MapReduce: Simplified Data Processing on Large Clusters* [online]. [cit. 2018-03-31]. Dostupné z: https://static.googleusercontent.com/media/research.google.com/cs//archive/mapreduce-osdi04.pdf

DUMBILL, Edd. *Big Data Now: 2012 Edition* [online]. 2012 ed. Sebastopol, CA: O'Reilly Media, 2012, s. 3–17 [cit. 2018-03-31]. ISBN 978-1-449-35671-2. Dostupné z: http://www.oreilly.com/data/free/files/big-data-now-2012.pdf

ELBATTAH, Mahmoud, Mohamed ROUSHDY, Mostafa AREF a Abdel-Badeeh M. SALEM. Large-Scale Entity Clustering Based on Structural Similarities within Knowledge Graphs. In: SOMANI, Arun K. a Ganesh Chandra DEKA, eds. *Big Data Analytics: Tools and Technology for Effective Planning*. Boca Raton, 2018, s. 312–332. ISBN 978-1-138-03239-2.

GIS Tools for Hadoop by Esri [online]. 2015 [cit. 2018-04-30]. Dostupné z: http://esri.github.io/gis-tools-for-hadoop/

GULLER, Mohammed. Big Data Analytics with Spark: A Practitioner's Guide to Using Spark for Large Scale Data Analysis. New York, New York: Apress, 2015. ISBN 978-1-4842-0965-3.

HOLUBOVÁ, Irena, Jiří KOSEK, Karel MINAŘÍK a David NOVÁK, 2015. *Big Data a NoSQL databáze*. První vydání. Praha: Grada. Profesionál. ISBN 9788024754666.

Infographic: The Four V's of Big Data. *IBM Big Data & Analytics Hub* [online]. 2013 [cit. 2018-03-13]. Dostupné z: http://www.ibmbigdatahub.com/infographic/four-vs-big-data

JEŽEK, Jan, Karel JEDLIČKA, Tomáš MILDORF, Jáchym KELLAR a Daniel BERA. Design and Evaluation of WebGL-Based Heat Map Visualization for Big Point Data. In: IVAN, Igor, Alex SINGLETON, Jiří HORÁK a Tomáš INSPEKTOR, eds. *The Rise of Big Spatial Data*. Springer, 2017, s. 13–26. ISBN 978-3-319-45122-0. ISSN 1863-2246.

KIM, Dale. MapR-FS vs. HDFS: The 5-Minute Guide to Understanding Their Differences – Whiteboard Walkthrough | MapR. *The Only Converged Data Platform | MapR* [online]. June 01, 2016 [cit. 2018-04-10]. Dostupné z: https://mapr.com/blog/mapr-fs-vs-hdfs-5-minute-guide-understanding-their-differences-whiteboard-walkthrough/

KLAUDA, Petr. Prostorově určená statistická data. *Statistika&My* [online]. Praha: Český statistický úřad, 2016, **6**(5), 18–19 [cit. 2018-04-30]. ISSN 1804-7149. Dostupné z: http://www.statistikaamy.cz/wp-content/uploads/2016/05/18041605.pdf

LANEY, Doug. *Deja VVVu: Others Claiming Gartner's Construct for Big Data* [online]. 2012 [cit. 2018-04-30]. Dostupné z: http://blogs.gartner.com/doug-laney/deja-vvvue-others-claiming-gartners-volume-velocity-variety-construct-for-big-data/

Leaflet - a JavaScript library for interactive maps [online]. 2018 [cit. 2018-04-30]. Dostupné z: http://leafletjs.com/

LEONE, Mike a Domenic AMATO. *Analyzing the Performance of MapR-DB, a NoSQL Database in the MapR Converged Data Platform* [online]. 2017 [cit. 2018-04-11]. Dostupné z: https://mapr.com/whitepapers/mike-leone-esg-lab-nosql-benchmark/assets/esg-lab-review-mapr-db-20170824.pdf

LIANG, Steve H. L. a Chih-Yuan HUANG. Geospatial Cyberinfrastructure for Addressing the Big Data Challenges on the Worldwide Sensor Web. In: KARIMI, Hassan A., ed. *Big Data: Techniques and Technologies in Geoinformatics*. Boca Raton: CRC Press, Taylor & Francis Group, 2014, s. 261–277. ISBN 978-1-4665-8651-2.

MORSTATTER, Fred, Jurgen PFEFFER, Huan LIU a Kathleen M. CARLEY. *Is the Sample Good Enough? Comparing Data from Twitter's Streaming API with Twitter's Firehose* [online]. 2013 [cit. 2018-04-27]. Dostupné z: http://www.public.asu.edu/~fmorstat/paperpdfs/icwsm2013.pdf

NYC Taxi & Limousine Commission [online]. 2018 [cit. 2018-04-26]. Dostupné z: http://www.nyc.gov/html/tlc/html/home/home.shtml

OpenLayers - Welcome [online]. 2018 [cit. 2018-04-30]. Dostupné z: http://openlayers.org/

OpenStreetMap Wiki [online]. 2018 [cit. 2018-04-30]. Dostupné z: https://wiki.openstreetmap.org/wiki/Planet.osm

OTTE, Adéla. *In-door analýza pohybu*. Brno, 2015. Bakalářská práce. Masarykova univerzita, Fakulta informatiky.

Portál AOPK ČR [online]. 2018 [cit. 2018-04-26]. Dostupné z: http://portal.nature.cz/

PruneCluster: An Improved Approach to Marker Clustering in Web-Based Mapping Services. *SINTEF* [online]. 2018 [cit. 2018-05-01]. Dostupné z: https://www.sintef.no/en/publications/publication/?pubid=CRIStin+1222327

RFC 7946 - The GeoJSON Format, 2016. *IETF Tools* [online]. [cit. 2018-04-30]. Dostupné z: https://tools.ietf.org/html/rfc7946

RŮŽIČKA, Jan, Lukáš ORČÍK, Kateřina RŮŽIČKOVÁ a Juraj KISZTNER. Processing LIDAR Data with Apache Hadoop. In: IVAN, Igor, Alex SINGLETON, Jiří HORÁK a Tomáš INSPEKTOR, eds. *The Rise of Big Spatial Data*. Springer, 2017, s. 351–358. ISBN 978-3-319-45122-0. ISSN 1863-2246.

Slovník VÚGTK. *VÚGTK, v.v.i.* [online]. 2018 [cit. 2018-04-19]. Dostupné z: http://www.vugtk.cz/slovnik/termin.php?jazykova_verze=&tid=5568&l=lidar

TRAME, Johannes a Carsten KEßLER. *Exploring the Lineage of VolunteeredGeographic Information with Heat Maps* [online]. 2011 [cit. 2018-05-02]. Dostupné z: https://www.researchgate.net/publication/216825402_Exploring_the_Lineage_of_Volunteered_Geographic_Information_with_Heat_Maps

Twitter Developer Platform [online]. 2018 [cit. 2018-05-01]. Dostupné z: https://developer.twitter.com/

VATSAVAI, Ranga Raju, Auroop GANGULY, Varun CHANDOLA, Anthony STEFANIDIS, Scott KLASKY a Shashi SHEKHAR. Spatiotemporal Data Mining in the Era of Big Spatial Data: Algorithms and Applications. In: *Proceedings of the 1st ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data - BigSpatial '12* [online]. New York, USA: ACM Press, 2012, 2012, s. 1–10 [cit. 2018-03-14]. DOI: 10.1145/2447481.2447482. ISBN 9781450316927. Dostupné z: http://academia.edu/2494125/Spatiotemporal_Data_Mining_in_the_Era_of_Big_Spatial_Data_Algorithms_and_Applications

Vektorová data — Školení Úvod do (Open Source) GIS, 2017. *GISMentors | Školení Open Source GIS* [online]. [cit. 2018-04-30]. Dostupné z: http://training.gismentors.eu/open-source-gis/formaty/vektor.html

WULDER, Michael A., Joanne C. WHITE, Thomas R. LOVELAND, et al. The global Landsat archive: Status, consolidation, and direction. *Remote Sensing of Environment* [online]. 2016, (185) [cit. 2018-03-14]. ISSN 0034-4257. Dostupné z: https://landsat.usgs.gov/sites/default/files/documents/1-s2.0-S0034425715302194-main.pdf

PŘÍLOHY

SEZNAM PŘÍLOH

Vázané přílohy:

Příloha 1 Časová náročnost agregace do čtvercové sítě

Volné přílohy

Příloha 2 Postup vytvoření virtuálního počítače s HDP Sandbox na platformě Microsoft Azure

Příloha 3 Licenční smlouva o poskytnutí dat AOPK ČR

Příloha 4 Poster

Příloha 5 DVD

Popis struktury DVD

Adresáře:

metadata

text_prace – obsahuje text práce ve formátu PDF

data

aopk_nalezy_geojson – obsahuje testovací vzorky dat ve formátu GeoJSON

aopk_nalezy_json – obsahuje testovací vzorky dat ve formátu JSON

tripdata_hdfs_export – obsahuje neupravená data exportovaná z HDFS

nyc_tlc_trip_data.gdb – obsahuje výsledné vrstvy, které vznikly převedením a sloučením exportovaných JSON souborů

geoprocessing_tools_for_hadoop – obsahuje toolbox pro propojení ArcMap a Hadoop

ndop_export.zip – export z nálezové databáze ochrany přírody

yellow_tripdata_2009-01.csv – taxi záznamy z ledna roku 2009

nalezy.mbtiles – vytvořené vektorové dlaždice

web – obsahuje webové stránky vytvořené v rámci bakalářské práce

poster – obsahuje poster vytvořený v rámci bakalářské práce ve formátu PDF

Export z nálezové databáze ochrany přírody je vázán licenční smlouvou uzavřenou s AOPK ČR. Další využití je možné jen se souhlasem správce těchto dat.

Tabulka 1 Časová náročnost zpracování CSV souboru o velikosti 2,5 GB na konfiguraci DS2_V3 (čas uveden v sekundách)

Měření	**Rozlišení**				
	0.01	**0.005**	**0.001**	**0.0005**	**0.0001**
1.	105,111	108,894	115,005	125,465	176,361
2.	106,109	110,560	115,957	128,980	179,137
3.	105,400	109,075	116,857	125,076	171,118
4.	105,929	112,027	115,932	126,701	173,548
5.	107,087	111,507	116,449	127,760	171,493
Aritmetický průměr	105,927	110,413	116,040	126,796	174,331
Medián	105,929	110,560	115,957	126,701	173,548

Tabulka 2 Časová náročnost zpracování CSV souboru o velikosti 2,5 GB na konfiguraci DS4_V2 (čas uveden v sekundách)

Měření	**Rozlišení**				
	0.01	**0.005**	**0.001**	**0.0005**	**0.0001**
1.	39,813	45,664	45,955	47,769	67,146
2.	38,964	44,638	46,655	48,908	73,920
3.	40,294	44,222	46,044	49,852	71,293
4.	40,014	45,380	45,662	47,969	67,134
5.	39,312	44,357	46,509	48,898	72,217
Aritmetický průměr	39,679	44,852	46,165	48,679	70,342
Medián	39,813	44,638	46,044	48,898	71,293

Tabulka 3 Časová náročnost zpracování CSV souboru o velikosti 10 GB na konfiguraci DS2_V3 (čas uveden v sekundách)

Měření	**Rozlišení**				
	0.01	**0.005**	**0.001**	**0.0005**	**0.0001**
1.	372,787	383,926	399,158	507,235	557,967
2.	371,778	379,790	402,686	503,575	562,649
3.	370,452	381,810	404,928	505,200	558,157
4.	372,373	379,961	401,005	508,053	562,512
5.	369,374	382,535	399,592	505,438	565,360
Aritmetický průměr	371,353	381,604	401,474	505,900	561,329
Medián	371,778	381,810	401,005	505,438	562,512

Tabulka 4 Časová náročnost zpracování CSV souboru o velikosti 10 GB na konfiguraci DS4_V2 (čas uveden v sekundách)

Měření	**Rozlišení**				
	0.01	**0.005**	**0.001**	**0.0005**	**0.0001**
1.	138,767	142,171	149,489	223,885	250,729
2.	137,552	145,879	149,209	220,579	251,696
3.	135,516	145,050	150,475	220,315	248,340
4.	138,517	142,663	152,739	221,452	250,220
5.	138,681	144,741	149,664	222,521	252,969
Aritmetický průměr	137,807	144,101	150,315	221,750	250,791
Medián	138,517	144,741	149,664	221,452	250,729