

**Vizualizace dopravních dat s
využitím prostorové databáze**
**Visualization of Traffic Data Using
Spatial Database**

Zadání diplomové práce

Student: **Bc. Vít Ptošek**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2612T025 Informatika a výpočetní technika

Téma: **Vizualizace dopravních dat s využitím prostorové databáze**
Visualization of Traffic Data Using Spatila Database

Zásady pro vypracování:

Účelem práce je zpracovat a vizualizovat dopravní data, která budou uložena v prostorové databázi. Data představují záznam pohybu vozidel po silniční síti, případně dopravní ukazatele jako je intenzita dopravy apod. Vizualizace těchto dat bude sloužit k přehlednému zobrazení výsledků analýz nad těmito daty.

Zásady pro vypracování:

1. Seznamte se s dopravními daty a způsoby jejich vizualizace v prostoru.
2. Popište vhodné nástroje a algoritmy pro vizualizaci dopravních analýz.
3. Navrhněte postup předzpracování dat tak, aby bylo možné analyzovaná data vizualizovat vhodným způsobem.
4. Navrhněte a implementujte algoritmy pro vizualizaci.
5. Proveďte testy algoritmů a to včetně testů výkonnostních.
6. Proveďte vyhodnocení dosažených výsledků.

Seznam doporučené odborné literatury:

- [1] Girardin, F.: Visualising the Pulse of the City, <http://www.infovis.net/printMag.php?lang=2&num=190>
- [2] <http://senseable.mit.edu/realtimerome/>
- [3] http://www.itoworld.com/static/mapping_and_spatial_analysis/ito_map.html
- [4] <http://www.fastcompany.com/1277918/information-visualization-next-frontier-design>

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **RNDr. Daniela Szturcová, Ph.D.**

Datum zadání: 01.09.2013

Datum odevzdání: 07.05.2014




doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Souhlasím se zveřejněním této diplomové práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v magisterských programech VŠB-TU Ostrava.

V Ostravě 16. dubna 2014



.....

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 16. dubna 2014



.....

Rád bych na tomto místě poděkoval své vedoucí diplomové práce paní RNDr. Daniele Szturcové, Ph.D. za cenné rady a připomínky. Dále své rodině a přátelům, kteří mě po dobu studií podporovali.

Abstrakt

Tato diplomová práce pojednává o zpracování a vizualizaci dopravních dat, která představují pohyb vozidel po komunikacích a jsou uložena v prostorové databázi. V rámci práce je popsáno několik nástrojů a algoritmů zaměřujících se na zpracování prostorových dat a jejich následnou vizualizaci. Součástí práce byl návrh a realizace aplikace sloužící k provádění výkonostních testů nad prostorovými dotazy a uskutečnění těchto testů. Cílem byla implementace nástroje pro vizualizaci prostorových dat z oblasti dopravy.

Klíčová slova: vizualizace, prostorová data, dopravní data, GIS, Oracle, .NET, DotSpatial, WGS-84

Abstract

This diploma thesis deals with processing and visualization of traffic data representing a vehicle movement on road networks using spatial database. In the framework of the dissertation some of tools and algorithms focusing on spatial data processing and visualization were described. The part of the thesis was to design and implement an application for spatial queries performance testing followed by performing the tests itself. The aim was to implement a tool for traffic spatial data visualization.

Keywords: visualization, spatial data, traffic data, GIS, Oracle, .NET, DotSpatial, WGS-84

Seznam použitých zkratk a symbolů

AJAX	– Asynchronous JavaScript and XML
ASP	– Active Server Pages
ČÚZK	– Český úřad zeměměřický a katastrální
DMS	– Degrees minutes seconds, převod souřadnic na stupně
GIS	– geografický informační systém (IS rozšířený o prostorovou složku)
GPS	– Global Positioning System
KML	– Keyhole Markup Language
MBR	– Minimum Bounding Rectangle
SDI	– spatial data infrastructure, infrastruktura prostorových dat
SPS	– Standard Positioning Service
TIN	– Triangulated irregular network, nepravidelná trojúhelníková síť
UTM	– Universal Transverse Mercator
VTOPÚ	– Vojenský topografický ústav
WGS84	– World Geodetic System

Obsah

1	Úvod	6
2	Prostorová data	7
2.1	Dopravní prostorová data	7
2.2	Požizování a získávání prostorových dat	7
2.3	GPS	8
2.4	Služby GPS	10
3	Typy prostorových dat	11
3.1	Rastrový datový model	11
3.2	Vektorový datový model	12
3.3	Srovnání datových modelů	16
3.4	Shapefile	16
4	Souřadnicové systémy	17
4.1	Mapová projekce	19
4.2	Latitude & Longitude	19
5	Prostorová databáze	21
5.1	Oracle Spatial	21
5.2	Oracle indexování	21
5.3	Oracle partitioning	23
5.4	Oracle prostorové dotazy	23
6	Vizualizace prostorových dat	27
6.1	Metody a algoritmy pro vizualizaci prostorových dat	27
6.2	Algoritmy	27
6.3	Kd-tree	28
6.4	Problém s vizualizací	28
6.5	Metody pro vykreslování překrývajících se linií	29
6.6	Hermite Interpolation	29
6.7	Frameworky a knihovny	34
7	Postup prací	35
7.1	Oracle Server a SQL Developer	35
7.2	Použití Google Maps	35
7.3	Benchmark	37
8	Testování	39
8.1	Typy testů	39
8.2	Poznámka k vydání aplikace a případná zjištění	41
8.3	Assembly dependencies	43
8.4	Výsledky testů	43

8.5	•	43
8.6	Bez prostorového indexu	44
8.7	Grafy	44
8.8	Textový výpis	47
9	Nástroj pro vizualizaci dopravních dat	51
9.1	Co je naimplementováno	51
9.2	Načítání dat do mapy	52
9.3	Určení Latitude a Longitude na mapě	52
9.4	Využití grafů	52
9.5	Na co je třeba si dávat pozor	55
10	Závěr	56

Seznam tabulek

Seznam obrázků

1	Reprezentace geoprvků ve vektorovém a rastrovém modelu.	8
2	Konstelace GPS satelitů	9
3	Rastrový model	12
4	Vektorový model	13
5	QuadTree - Čtyřstrom	14
6	Základní typy geoprvků	14
7	Využití vrstev u vektorového a rastrového modelu	15
8	Zobrazení před reprojekcí z S-JTSK do WGS84	18
9	Zobrazení po reprojekci z S-JTSK do WGS84	19
10	Projekce 3D Země na Mercator	20
11	Obálka prvku	22
12	R-tree	24
13	Oracle segmentace tabulky	25
14	Skladba některých binárně-topologických vztahů při prostorovém dotazování	26
15	Kd-tree	30
16	Pseudokód pro vytvoření dvojdimenzionálního Kd-stromu	31
17	Druhy vykreslování překrývajících se linií	31
18	Funkce na Hermitovskou interpolaci	32
19	Před použitím Hermitovské interpolace	33
20	Po použití Hermitovské interpolace	33
21	Vykreslení bodů z databáze do Google Maps	36
22	Alter table a update	38
23	První typ testu	40
24	Druhý typ testu	40
25	Třetí typ testu	40
26	Čtvrtý typ testu	40
27	Test spuštěn z příkazové řádky nebo nástroje pro vizualizaci dopravních dat – průběh	41
28	Test spuštěn z příkazové řádky nebo nástroje pro vizualizaci dopravních dat – konec	42
29	Benchmark.exe – výpočet rozmezí a start testu	42
30	Zahájení testu a uložení vygenerovaných souborů	43
31	Assembly dependencies aplikace Benchmark	44
32	45
33	46
34	46
35	Graf třetího testu bez indexu, počet dotazů 16	47
36	Graf třetího testu bez indexu, počet dotazů 1600	48
37	Graf třetího testu s indexem, počet dotazů 16	48
38	Graf třetího testu s indexem, počet dotazů 1600	49
39	Textový výpis výsledku třetího testu s indexem, počet dotazů 1600	50

40	Výpis základních informací o zvoleném segmentu	52
41	přidání shapefilu do mapy jako vrstvu ve skupině Custom layers	53
42	Možné zobrazení České Republiky za pomoci přidanych vrstev	54
43	54
44	54
45	závislost rychlosti nákladních vozů na jejich množství	55

1 Úvod

V současné době se geografické informační systémy stále víc a víc dostávají do podvědomí uživatelů a těší se oblibě především díky přehlednosti zpracovaných dat a funkcí, které mnohdy zdarma nabízejí. Nejinak je tomu u GIS poskytujících služby zaměřené na dopravní informace. Se zvyšujícím se zájmem o využívání těchto služeb založených na geografických datech stoupá i náročnost požadavků uživatelů na ně. Málokdo dnes používá klasické papírové mapy namísto satelitních navigací nebo online map. Vývoj interaktivních map s aktualizovanými daty urazil velkou cestu, přesto je však stále v tomto směru co zlepšovat.

Aby bylo možné takové služby poskytovat, je nutné najít způsob, jak získané dopravní prostorová data analyzovat, zpracovávat a vizualizovat. Existuje několik algoritmů, díky nimž lze tohoto úkolu dosáhnout.

V teoretické části jsem se pokusil některé z nich popsat a vysvětlit jak fungují, jaký problém řeší a v čem lze nalézt jejich využití. V praktické části jsem určité z nich použil k implementaci nástroje pro vizualizaci dopravních dat za použití prostorové databáze.

2 Prostorová data

Tato data jsou někdy označována za geodata, jelikož obsahují prostorovou složku, tedy data vztahující se ke specifickým místům v prostoru, u kterých na potřebné úrovni známe lokalizaci. Pod pojmem prostorová data si můžeme představit jakoukoliv informaci ohledně umístění a tvaru fyzického objektu spolu s jeho geografickými vlastnostmi, což lze vnímat jako složku, která data váže k určitému místu na Zemi.

Prostorová data, která považujeme za pro nás zajímavou část reálného světa, je zapotřebí nějakým způsobem reprezentovat. Zásadní je přístup k informacím o geoprvcích, který může být objektový (sledujeme fyzické nebo abstraktní objekty), jevový (kvalitativní nebo kvantitativní vlastnost reálného světa v prostorových dimenzích jevu) a procesní (aktivity ovlivňující objekty, jevy a procesy reálného světa). Tyto prvky pak nesou informace prostorové (poloha, tvar, topologie), popisné (atributy a jejich hodnoty) a časové.

Reprezentace prostorových dat spočívá v převedení informací reálného světa v prostorovém modelu (rastrový nebo vektorový) do digitální podoby.

Prostor můžeme rozdělit a vyjádřit v několika dimenzích vzhledem ke studovanému jevu na 0 rozměrný - bodový, 1D – liniový, 2D – plošně rovinný, 2.5D – plošně prostorový a 3D – objemový, kdy použití každého typu prostorové dimenze záleží na účelu použití.

2.1 Dopravní prostorová data

Jestliže je řeč o dopravních prostorových datech, mohou taková poskytovat informaci o dění a situaci dopravy na určitém úseku dopravních komunikací jako je například rychlost, počet vozidel, jejich pohyb a plynulost na konkrétním úseku v reálném čase. Z takových informací pak lze těžit na poli statistického zpracování, což vede k zjištění korelace, závislosti záznamů a možné predikce vývoje situace v budoucnu. Statistickým zpracováním se dnes zabývají i společnosti IBM se svým produktem Smarter Traveler - Traffic Prediction Software a TomTom s aplikací Traffic. Google disponuje dostupnou aplikací Google Traffic.

2.2 Pořizování a získávání prostorových dat

Postupem času je možno tato data získávat z rozličných zdrojů podle dostupnosti. Dnes existují jak zdroje veřejné, tak i ty nejrůznějších komerčních organizací. Oba tyto zdroje mohou poskytovat data primárního a sekundárního charakteru, přičemž sledujeme, co a jakým způsobem je sbíráno. Zástupci veřejných zdrojů jsou kupříkladu ČÚZK a VTOPÚ.

Primární zdroje dat většinou vznikají sběrem výzkumných týmů, případně privátních firem (geodeti, zeměměřiči, firmy sledující bezpečnost vozidel apod.), měřením v terénu, dálkovým či jiným pozorováním Země, dotazníkovým průzkumem nebo laboratorními pokusy. Takový sběr dat může probíhat jako zaznamenávání a odesílání svého pohybu po určité trase za použití GPS zařízení. Výsledkem jsou pak, mimo jiné, souřadnice a názvy objektů spolu s časem pořízení a doplňujícími informacemi.

Sekundárními zdroji dat rozumíme informace o získání dat jako takových, postupech zpracování, jejich kvalitě a aktuálnosti – lze je nazývat metadaty.

Reprezentace geoprvků v modelech



Obrázek 1: Reprezentace geoprvků ve vektorovém a rastrovém modelu.

Při pořizování geodat mohou nastat chyby způsobené pořizováním, transformacemi dat nebo jejich analýzou či interpretací výsledků. Tyto chyby je třeba zdokumentovat a zohlednit jejich vliv na výsledná data.

Detekce chyb může být prováděna na několika složkách – grafické, tématické a časové

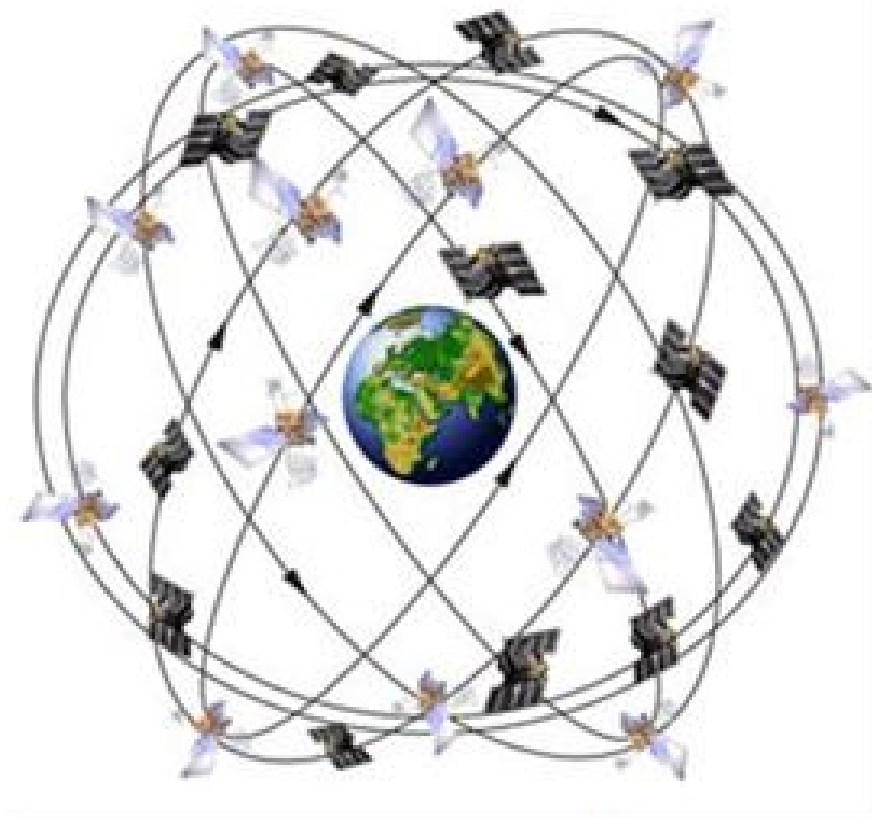
2.3 GPS

Jedná se o vojenský globální družicový polohový systém, který je následníkem GNSS Transit a je vylepšen o kvalitu a přesnost dostupných služeb. Někdy je možné jej nalézt pod původním jménem NAVSTAR GPS, který dnes zůstal pouze družicím využívaným pro službu GPS.

Systém je pod záštitou Ministerstva obrany Spojených států amerických a lze díky němu kdykoliv zaměřit přesnou polohu na Zemi spolu s přesným časem. Původně byla přesnost schválně zkreslena kvůli bezpečnostním podmínkám, jelikož panovala obava ze zneužití k teroristickým útokům. V současné době je možno zaměřit lokaci s přesností až do deseti metrů. Tuto přesnost lze zvýšit za pomoci různých metod na jednotky v řádech centimetrů.

GPS se skládá ze třech segmentů:

- Kosmický
- Řídící a kontrolní
- Uživatelský



Obrázek 2: Konstelace GPS satelitů .

2.3.1 Kosmický segment

Skládá se z konstelace alespoň 24 funkčních satelitů (95% času od vzniku své existence v počátcích 80. let 19. století), které vysílají rádiové signály uživatelům. Tyto družice jsou umístěny ve výšce 20200 km na 6 kruhových drahách, při čemž každá z nich oběhne zeměkoulí dvakrát denně. Pokud je funkčních satelitů méně, nemluvíme o plné operační schopnosti, ale pouze o schopnosti částečné operační.

2.3.2 Kontrolní segment

Tento segment se skládá z částí, jakými jsou velitelství a řídicí středisko, dále z tří povělových stanic a 18 stanic monitorovacích. Slouží k monitorování kosmického segmentu a zasilání povelů družicím. Taktéž zajišťuje jejich manévry a správu atomových hodin, jež jsou nezbytné pro přesný čas. Pokud by toto nebylo z nějakého důvodu možné, na scénu

nastupuje režim AUTONAV, díky čemuž jsou družice teoreticky schopny fungovat ještě následujícího půl roku. Jelikož zatím nebyla potřeba uvést tento mód v aktivní, nemohlo být provedeno žádné testování funkčnosti.

2.3.3 Uživatelský segment

Uživatelský segment se pro koncového uživatele jeví jako nejzásadnější, jelikož jej přes signály viditelných družic používá za pomoci GPS přijímače, tedy je základem pro odchyt potřebných dat. Pokud jsou k dispozici alespoň čtyři družice, které jsou v danou chvíli nad obzorem, lze obdržet polohu přijímače, jeho nadmořskou výšku a přesné datum s časem. Tato komunikace je jednosměrná od družice k GPS zařízení.

Co se přesnosti týče, je zde třeba rozlišit typ uživatelů na autorizované (vojenský sektor) a ostatní (zde se řadí především civilní sektor). Vláda USA poskytuje GPS služby v úrovních specifikovaných ve standardu, které jsou popsány v SPS, kdy v nejhrošším případě slibuje akurátnost na 7,8 metrů. Přesnost pro civilní i vojenské použití je naprosto stejná s tím rozdílem, že v prvním případě je použito jedno pásmo, kdežto v druhém případě jsou použity pásma hned dvě, což má za následek určité výhody, poněvadž dochází k ionosférické korekci a tím pádem i k zpřesnění. Stav ionosféry ovlivňuje i případná cyklická aktivita Slunce, která se děje pravidelně každých 11 let, případně pak meteorologické vlivy.

Oba tyto typy autorizací používají referenční systém WGS 84.

2.4 Služby GPS

Jsou volně dostupné pro civilní účely a v případě USA i k vojenským účelům. Pro nás je důležité, že tuto službu můžeme využívat v navigacích, leteckých i pozemních dopravních systémech, sběru dat či například i k zábavě v podobě geocashingu za výše zmíněných podmínek a kvality. GPS podléhá neustálému vylepšování a modernizaci.

3 Typy prostorových dat

Jak již bylo zmíněno, prostorová data využívají dva základní datové modely. Reprezentace probíhá v rastrovém či vektorovém modelu. V obou případech ve svém prostorovém modelu odráží informace z reálného světa v digitální podobě, a to za pomoci bodů, linií a polygonů, které zastupují plochy. Například loďka na jezeře může být prezentována bodem, její trajektorie pohybu linií a jezero samotné polygonem.

3.1 Rastrový datový model

Rastrový model si můžeme představit jako rozdělení plochy nebo prostoru na jednotlivé stejně velké buňky, čímž nám vzniká pravidelná mřížka, jejíž každý jeden díl je atomický. Elementy, které rovinnou plochu zcela vyplňují, mohou mít různý tvar a velikost, při čemž zastupují objekty a jevy v 2D, 2.5D, nebo 3D poloze prostoru. Důležité je, aby tvar buněk měl schopnost nekonečného opakování se v prostoru, případně aby splňoval nekonečnou rekurzi na menší části o stejném tvaru (fraktály).

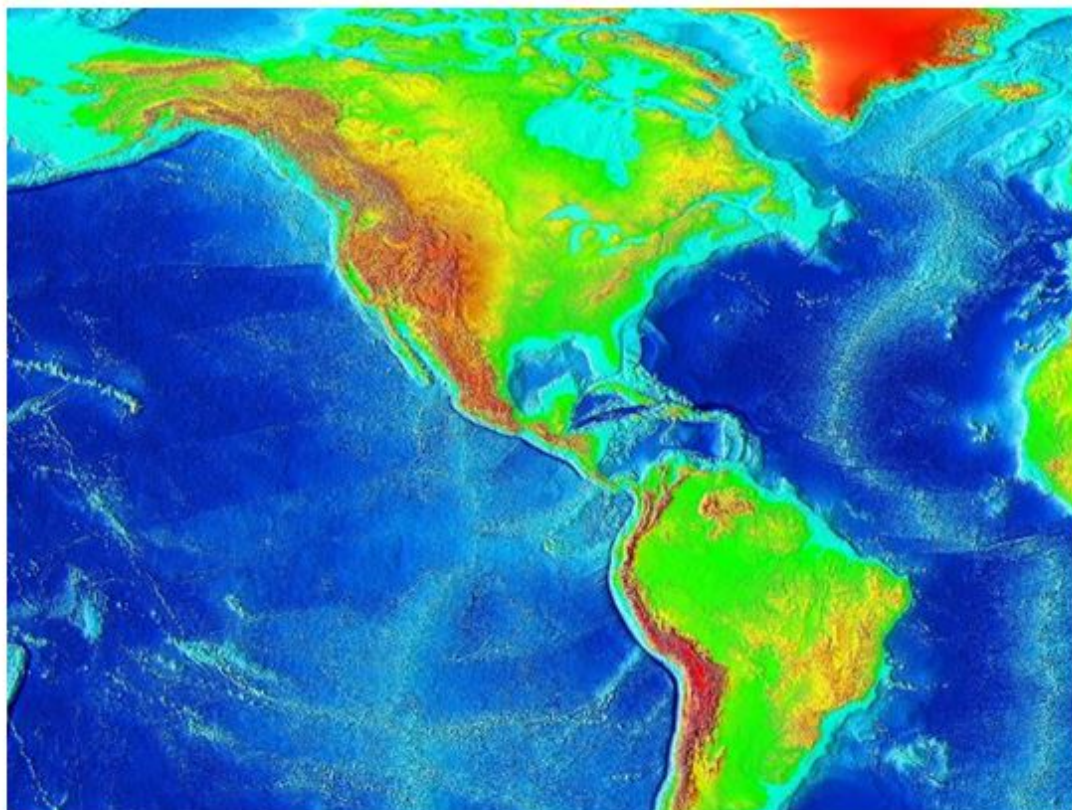
Prostorové vztahy jsou mezi geoprvky rastrového datového modelu dostupné již přímo v rastru, což neplatí u vztahů mezi prostorovými prvky a lokalizací. Takovýto prostorový element je základní informační jednotkou a jeho atributy se pojí s konkrétní buňkou.

Čím je jednotlivá buňka menší, tím se jich v řádcích a sloupcích objevuje více, což má za následek zvýšení rozlišení a tedy i kvality co do ostroty. Každá buňka obsahuje jedinečnou adresu skládající se ze sloupcového a řádkového indexu (poziční index) spolu s libovolnou hodnotou, případně dokonce hodnotou žádnou, v tom případě nese tzv. nulová data (NODATA). Pokud nějakou hodnotu má, pak v podobě mapovaného čísla nebo kódu. Číslování zpravidla postupuje od levého horního rohu k pravému dolnímu rohu.

Kvalita rastru může být kromě velikosti buněk způsobující rozlišení ovlivněna i hodnotou zobrazovaného atributu. Taková hodnota je pak například bodová, nebo ji získáme dopočítáním aritmetického průměru či váženého aritmetického průměru několika bodů nebo výběrem maximální/minimální hodnoty atributu. Vliv má i barevná hloubka rastru, kdy u binárních rastrů hodnoty nabývají například pouze nul a jedniček, přičemž u 24bitového rastru již rozlišujeme 2^{24} (16777216) různých celočíselných hodnot každého atributu.

U rastrového modelu hraje roli ještě další faktor, a to zda požadujeme pro každý dílek stejnou rozlišovací úroveň. Pokud tomu tak je, používají se stejně velké buňky. V opačném případě, pokud je zapotřebí různé rozlišovací úrovně, potažmo hierarchického členění, můžeme pozorovat, že se velikost buněk liší dle definice.

Prostor lze pravidelně či nepravidelně členit a dělit. U dělení záleží na tom, zda je buňka vždy stejného tvaru a velikosti (pravidelný prostor), nebo nikoliv (nepravidelný prostor). Nepravidelného dělení prostoru se využívá především u TIN, která pro reprezentaci povrchu používá množinu trojúhelníků za použití třech bodů kdekoliv v prostoru. Pravidelné členění využívá tvary buněk čtvercové (nejčastější díky shody s kartézským souřadnicovým systémem), trojúhelníkové a šestiúhelníkové (hexagonální mřížka, která



Obrázek 3: Rastrový model

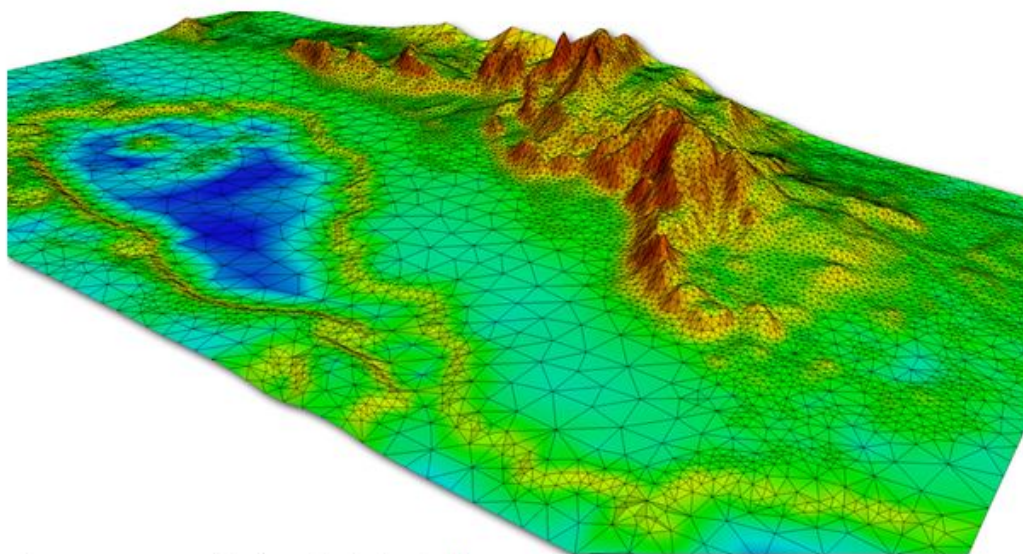
lehce připomíná plástve medu). Informace o buňkách jsou uloženy přímo v databázi (již zmíněný poziční index s hodnotou prvku).

Pokud chceme nad rastrovými soubory provádět některé z operací, musí mít tato data registrované vrstvy, být schopna převzorkování souboru při rozlišné velikosti buněk a transformace prostorových souřadnic prvku. Hojně se u rastrových údajů využívá komprese dat s cílem ušetřit místo na disku, aniž by došlo při uložení k zmenšení kapacity dat na úkor jejich kvality – ztrátě informace. Známé metody pro bezztrátovou kompresi dat jsou řádková a RLE (kdy například zápis 11122999999999 nahradíme za (3 1) (2 2) (9 9)), quadtree a Mormonovo pořadí.

3.2 Vektorový datový model

Vektorový datový model tvoří spolu s rastrovým modelem dva nejrozšířenější datové modely.

Od rastrového se liší v rámci reprezentace reálného světa ve využití schématického rozdělení geoprvků na tři jednotlivé složky, kterými jsou unikátní identifikátor oddělený



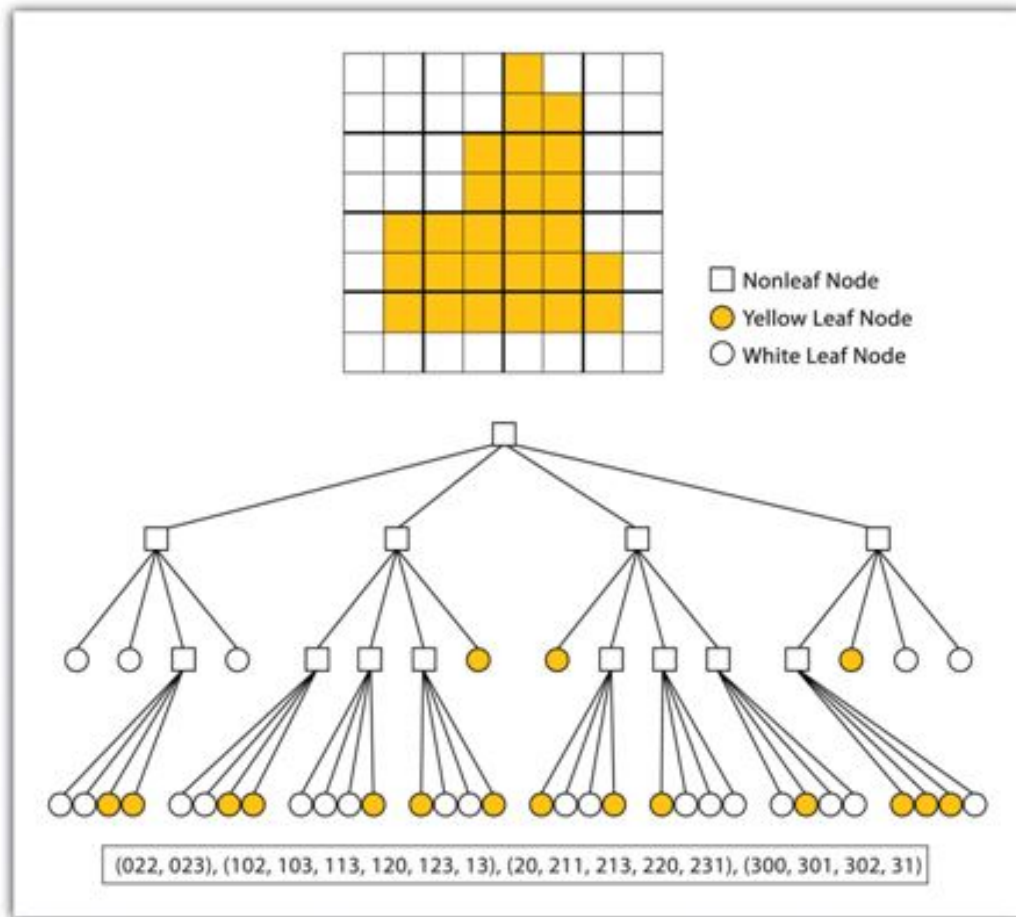
Obrázek 4: Vektorový model

od dvou dalších složek - geometrického popisu geoprvku a atributů geoprvku. Tento jedinečný identifikátor zároveň slouží jako pojítka mezi těmito dvěma složkami.

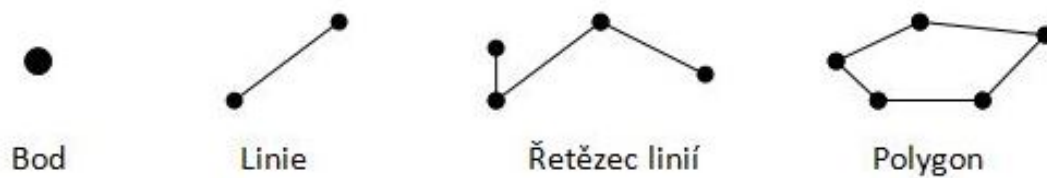
Samotnou reprezentaci geometrie vektorového modelu tvoří lineární geometrické prvky (vektory, případně sekvence vektorů), které jsou zároveň konečné, spjité a stejnorodé. Mohou být rovněž reprezentovány v různých dimenzích.

1. 0D – Vektor definovaný souřadnicemi v prostoru, který má totožný počáteční a koncový bod, a tudíž nabývá nulové délky, označujeme za bod.
2. 1D – Posloupnost vektorů (úseček), které spolu sousedí. Pokud uzel (hrana) úsečky spojuje počáteční a koncový bod (oba tyto uzly se musí lišit), jedná se o linii. Linie na sebe mohou navazovat a tvořit tak řetězec linií. Řetězce dělíme na otevřené (stále jedna dimenze) nebo na uzavřené, které již patří do 2D.
3. 2D – Uzavřený řetězec linií se nazývá plocha.
4. 2.5D – Plocha nadstavená o hodnotu v každém bodě reprezentuje povrch.
5. 3D – Prvek, u kterého lze měřit ve třech rozměrech zastupující objem.

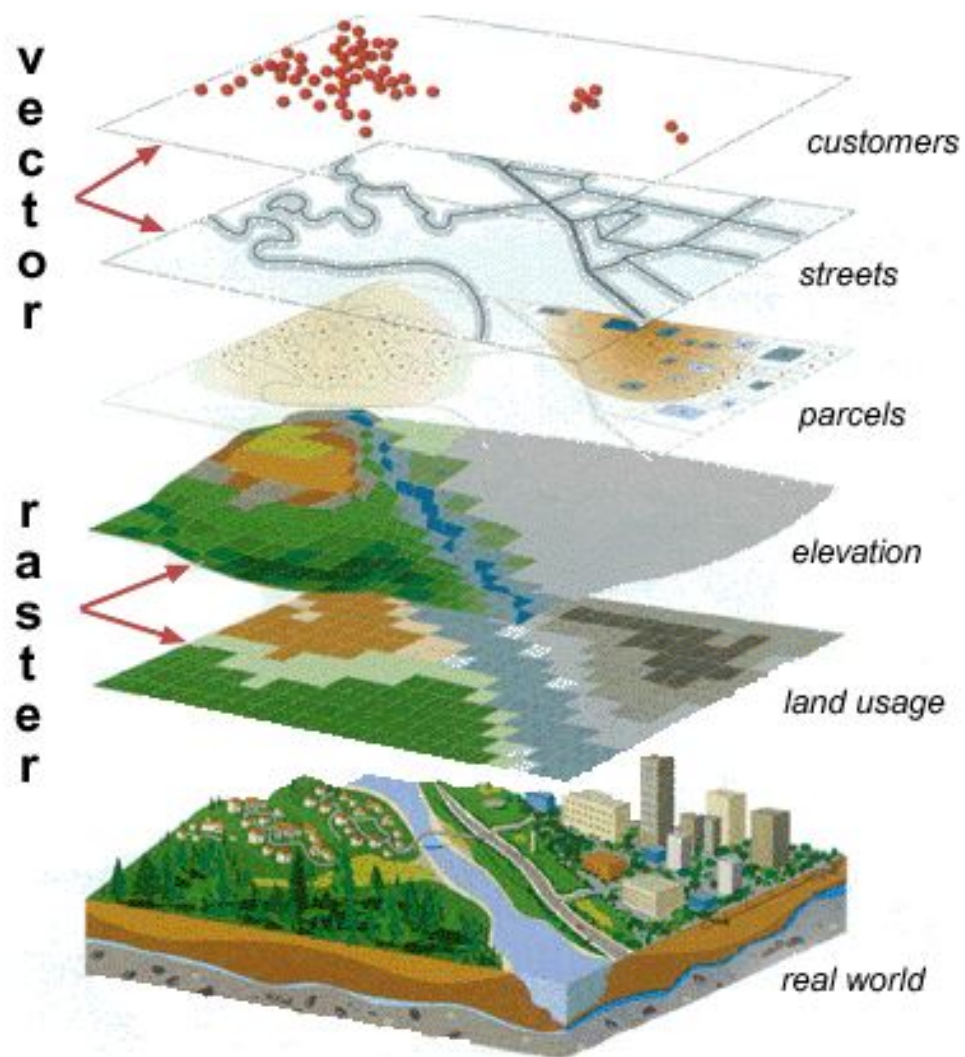
Vektorový datový model využívá topologických vztahů liniových a prostorových objektů. U topologie můžeme sledovat konektivitu, definování plochy a sousednost. Velikou výhodou jsou vrstvy, které hromadí prvky stejné geometrie. Nové geoobjekty se zpravidla skládají alespoň ze dvou vrstev. Díky vrstvám se zrychlilo vyhledávání podle atributu, je možné tvořit tematické hierarchie a přistupovat k prvkům pro každou vrstvu zvlášť.



Obrázek 5: QuadTree - Čtyřstrom



Obrázek 6: Základní typy geoprvků



Obrázek 7: Využití vrstev u vektorového a rastrového modelu

3.3 Srovnání datových modelů

Nedá se jednoznačně říct, který model je lepší. Oba mají své výhody a nevýhody a mohou být více či méně vhodné vzhledem ke konkrétnímu způsobu použití. Rastrový datový model přes odrazující větší objem dat a fakt, že velikost buňky a rozlišení ovlivňují přesnost, disponuje jednoduchostí datové struktury. Vektorový datový model vyniká v polohové přesnosti (přibližování mapy nemá vliv na vzhled vektorového prvku a dá se snadno převádět mezi souřadnicovými systémy) a nezabírá takový objem uložených údajů jako model rastrový, ovšem jeho datová struktura je komplikovanější a je časově náročnější kvůli složitým výpočtům při analýzách a vytváření topologie.

3.4 Shapefile

Shapefile je otevřený datový formát vyvinutý americkou firmou Esri se záměrem především ukládat vektorová prostorová data v geografických informačních systémech. Ukládání pracuje na bázi netopologické geometrie a atributové informace pro geoprvek v rámci jedné datové sady. Geometrie prvku se ukládá za pomoci vektorových souřadnic. Silná stránka tohoto datového formátu spočívá v rychlé vizualizaci geodat a možnosti editace bodů. Naopak za nevýhodu můžeme považovat redundanci dat (v případě bodů sousedících polygonů).

Slouží k prostorovému popisu geometrických bodů, linií a polygonů (ploch) spolu s jejich atributy, které daný prvek popisují. Shapefile rozeznává 14 typů geometrických prvků, kdy každý z nich má svůj specifický a jedinečný kód. V mapách jeden soubor obvykle zastupuje jeden a více objektů z reálného světa jakým je kraj, obec, les, vrstevnice, oceán apod. V každém shapefile však může být uložen pouze jeden typ geometrie (například pouze body nebo polygony).

Formát shapefile je definován minimálně třemi povinnými soubory, kdy každý z nich je velikostně omezen na 2 GB limit. Celkový součet však může být větší (například 3 soubory po 1 GB). Povinné soubory se skládají z hlavního souboru (*.shp), který ukládá geometrii prvků, indexového souboru (*.shx), jenž slouží k ukládání indexů pro geometrii prvků a souboru, čímž dojde k propojení prvku v hlavním souboru se záznamem v atributové tabulce. Posledním povinným souborem je soubor databázový (*.dbf), což je soubor tabulky dBASE s účelem ukládat atributové informace náležící prvkům – každý záznam v tabulce reflektuje jeden prvek.

Kromě povinných souborů je možné využít i souborů doplňkových, které slouží například pro uložení informací o souřadnicovém systému a projekci popsané prostým textem nebo specifikaci kódové stránky pro korektní identifikaci znaků.

4 Souřadnicové systémy

Někdy nazývané soustavou souřadnic či systémem souřadnic. Souřadnicový systém je souborem referenčních bodů, přímek a křivek, díky nimž je možné určit souřadnice polohy prvku v prostoru vzhledem k vztažné soustavě. Poloha prvku je udávána souřadnicemi (koordináty) prvku, které mohou zastávat vzdálenost nebo úhly vztažené k referenčním bodům a přímkám/křivkám vybrané souřadné soustavy, na které závisí význam souřadnic.

Pokud chceme určit polohu geoprvku, potřebujeme znát druh soustavy souřadnic, počátek soustavy (datum), směr a počet souřadných os a jednotky, ze kterých budeme vycházet (násobky a díly jsou výslednými hodnotami souřadnic). To znamená, že stejné místo na Zemi můžeme vyjádřit různými způsoby, což způsobí, že souřadnice se budou lišit.

Ve své diplomové práci jsem používal data dodaná ve světovém geodetickém referenčním systému 1984, který dále budu označovat pouze jako WGS84. Dále data volně dostupná na Internetu, která byla většinou v souřadnicovém systému S-JTSK. Prvně uvedený souřadnicový systém je světově uznávaný geodetický standard, který je neustále upravován dle nových měření a je hojně používán státy NATO. Jedná se o pravotočivou kartézskou soustavu souřadnic, kdy referenční plochou je elipsoid s počátkem v hmotném těžišti (středu) Země s přesností až 2 cm a používá kartografické zobrazení UTM. Osy X a Y leží v rovině rovníku. Osa Z vyjadřuje osu rotace Země v roce 1984. Počátek společně s orientací těchto os je realizován za pomoci 12 pozemských stanic, u kterých jsou známy jejich přesné souřadnice. Ty jsou neustále monitorovány dráhami družic systému GPS-NAVSTAR. Délka hlavní poloosy měří 6 378 137 metrů a délka vedlejší poloosy se rovná 6 356 752,3142 metrům.

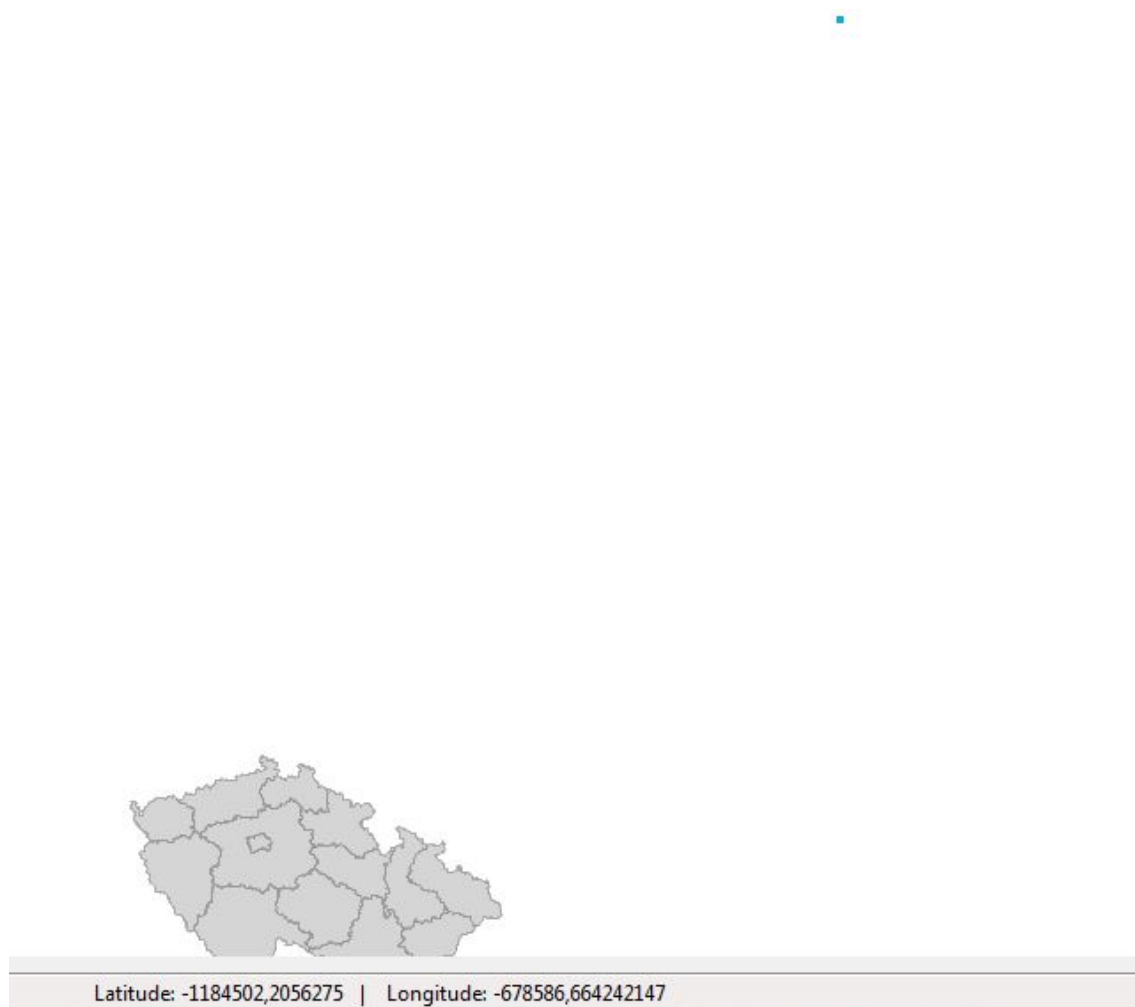
Souřadnicový systém WGS84 jsem si zvolil jako primární a rozhodl jsem se dále pracovat výhradně s ním. Pokud jsem pracoval s daty v jiném referenčním systému, musel jsem je nejprve přetransformovat právě do WGS84.

Na území České republiky se také hojně využívá souřadnicového systému Jednotné trigonometrické sítě katastrální (S-JTSK), který využívá Křovákovo dvojité konformní kuželové zobrazení v obecné poloze s výškopisem Československé jednotné nivelační sítě (dříve ve výškovém systému Jadranském, později se přešlo na výškový systém Baltský), a souřadnicového systému 1942 (S-42). Shapefiley veřejně dostupné a stažené z Internetu byly právě v tomto systému souřadnic.

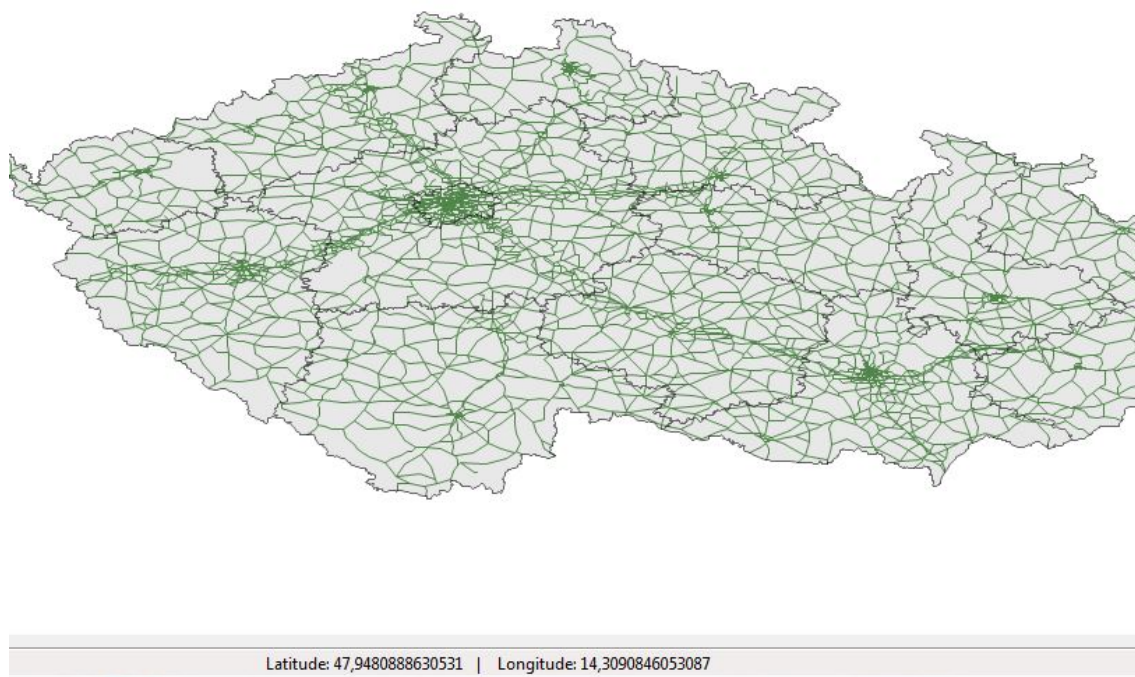
Na následujících dvou obrázcích je patrné, jak moc se souřadnicové systémy mohou lišit v měřítku (rozdíl ve velikosti je značný), lokalizaci (poloha České republiky je na dvou rozdílných místech) a v pootočení (krajina není natočena vodorovně). V obou případech jsou přidány stejné vrstvy (nahodile vygenerované body na území České republiky v WGS84 a okresy České republiky S-JTSK).

V prvním případě bylo třeba mapu značně oddálit, aby byly vidět obě vrstvy. Modrý bod nahoře zobrazuje silnice na území České republiky. Také souřadnice jsou v záporných číslech.

V druhém případě se již vrstvy správně překrývají, jelikož po transformaci zastupují stejnou oblast ve stejném souřadnicovém systému a souřadnice jsou kladné.



Obrázek 8: Zobrazení před reprojekcí z S-JTSK do WGS84



Obrázek 9: Zobrazení po reprojekci z S-JTSK do WGS84

4.1 Mapová projekce

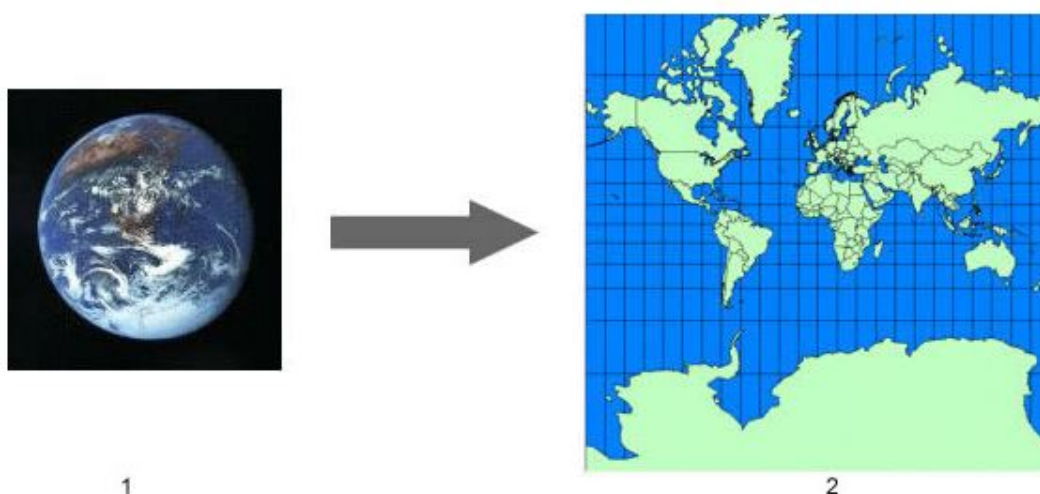
Mapová projekce neboli kartografická projekce značí matematické vyjádření, které slouží k reprezentaci zaobleného trojrozměrného zemského povrchu za pomoci ploché dvou-rozměrné mapy. Jelikož se převádí 3D mapa (případně model - glóbus) na 2D mapu (klasická mapa), musí nutně dojít k deformaci minimálně jednoho z aspektu mapy, což má za následek, že se neshoduje vlastnost mapy, jakou je měřítko, tvar nebo například plocha.

Dosud bylo vyvinuto na stovky projekcí, které se používají, aby co nejvěrněji zachytily realitu podle určitého prvku, který je potřebný pro určitý typ mapy. To vše však na úkor jiných vlastností, které tak potřebné nejsou.

Mezi v současnosti hojně využívané mapové projekce patří Mercatova projekce, která zachovává pravé úhly mezi rovnoběžkami a poledníky na úkor ploch. Lambertova konformní kuželová projekce zase na úkor plochy nabízí nejpřesnější měřítko.

4.2 Latitude & Longitude

Zeměpisné souřadnice neboli koordináty jsou nutností pro správné a jednoznačné určení polohy na povrchu Země. Jde o sférické souřadnice s počátkem ve středu Země, zeměpisnou šířku (anglicky latitude) a zeměpisnou délku (anglicky longitude).



Obrázek 10: Projekce 3D Země na Mercator

Zeměpisná šířka určuje polohu na povrchu Země směrem na sever nebo jih od rovníku. Je určena úhlem, který svírá rovina rovníku s normálou - přímkou kolmou na danou rovinu. Měří se ve stupních a lze ji specifikovat v úhlových minutách, vteřinách a jejich zlomcích. Jedna minuta zeměpisné šířky na povrchu Země odpovídá 1852 metřům.

Zeměpisná délka určuje polohu na povrchu Země směrem na východ nebo západ od nultého poledníku (též základní nebo Greenwichský poledník). Je určena úhlem svírajícím rovinu místního poledníku procházejícího určeným bodem a rovinu nultého poledníku. Stejně jako zeměpisnou šířku i zeměpisnou délku měříme ve stupních.

Za předpokladu, že máme obě tyto koordináty a zvolený souřadnicový systém, můžeme s přesností zaměřit a určit kterékoliv místo na povrchu planety Země, které souřadnicím a referenčnímu systému odpovídá. Souřadnice se pak zpravidla uvádí v decimálním nebo stupňovém (DMS) zápisu. Hodnoty v těchto zápisech lze obousměrně převádět.

Vždy nejdříve uvádíme zeměpisnou šířku následovanou zeměpisnou délkou. Například zápis Lat Long (49.874988, 18.437076) odpovídá po převedení na stupně zápisu DMS (49° 52' 29.9568" N, 18° 26' 13.4736" E). Dalším možným převodem je konvertovat z Lat Long či DMS na UTM (315857.24, 5527881.57) s UTM zónou 34U.

Časté problémy vznikají právě přehozením pořadí latitude a longitude, čímž je samozřejmě určena zcela jiná poloha (pokud obě hodnoty nejsou totožné). Jelikož jsem v práci používal souřadnicový systém WGS84, chvíli mi trvalo, než jsem tuto chybu odhalil. Jednou z výhod upřednostnění souřadnicového systému S-JTSK před WGS84 je fakt, že na území České republiky musí být vždy nutně osa X ζ Y (tedy latitude je větší číslo než longitude) a díky tomu se dá ošetřit správnost pořadí koordinátů.

Ve své práci používám výhradně zápis Lat Long namísto stupňů nebo UTM.

5 Prostorová databáze

Prostorový databázový systém (anglicky spatial database) je objektově-relační systém řízení báze dat schopný provádět správu nad prostorovými daty. K těmto datům je pak možno přistupovat a dále je zpracovávat, ukládat, dotazovat se nad nimi a celkově s nimi manipulovat.

Dnes již existuje zástup open sourcových, neplacených i placených databázových systémů s podporou geodat. Ve své práci jsem využíval prostorový databázový systém Oracle Spatial, který patří k těm nejrozšířenějším a nejkvalitnějším s optimálním výkonem.

5.1 Oracle Spatial

Oracle Spatial představuje nadstavbu k produktu Oracle Database, díky níž je umožněno použití i v těch nejnáročnějších geoinformačních systémech. Nabízí spoustu vnořených metod, funkcí a procedur, které uskutečňují načítání a ukládání geodat a pomáhají s následnou prací nad nimi.

Plná integrace tohoto volitelného doplňku v Oracle Database umožňuje plné využití všech součástí Oracle Server navýšené o prostorově založené záznamy, s kterými lze pracovat rychle a efektivně díky prostorové indexaci. K Oracle prostorové databázi se pojí i doplňky pro práci s geodaty, jakými jsou například Georaptor, Oracle Locator nebo nástroj MapViewer.

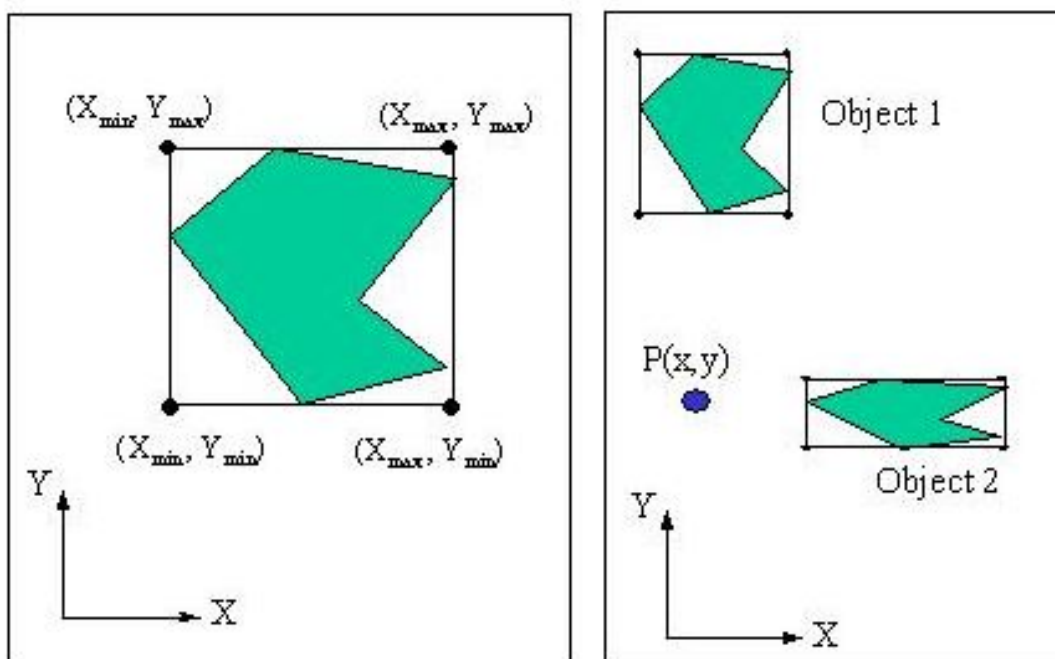
Se Spatial modulem se Oracle server rozšířil o některé vlastnosti pojící se k prostorovým a geometrickým prvkům jako je například řízení projekce a transformace, lineární referenční systém, analýza prostorových dat, syntaxe a sémantika zaměřená na ukládání geometrických dat a vykonávání prostorových dotazů spolu s utilitami na podporu práce s geoprvky.

Samozřejmostí je zavedení a podpora geometrických typů jakými jsou body, linie, řetězce linií, polygony, kružnice a oblouky. Aby s těmito typy mohl Oracle Spatial pracovat, zavedl si vlastní objektový datový typ SDO.GEOMETRY pro ukládání a prezentaci prostorových dat. U všech objektů je možné uvádět směr onoho objektu spolu s popisem a jinými atributy prvku jako jeden záznam v jedné tabulce, a to i pod různými referenčními systémy. Samotná geometrie prvku je zastoupena záznamem v jednom řádku a sloupci, kdy obsahuje vše potřebné v jedné buňce – celá geometrická složka je tedy uložena v jediném atributu prostorové tabulky. Z tohoto atributu je možno konkrétní záznamy extrahovat (typ prvku, souřadnice apod.). Lokalizační údaje se modelují na vrstvách tabulky se stejným souřadnicovým systémem.

5.2 Oracle indexování

Indexy jsou nepovinné struktury pojící se k určité tabulce nebo clusteru, které slouží k zrychlení přístupu k záznamům asociované tabulky. Díky nim odpadá nutnost procházet tabulku sekvenčně, tedy záznam po záznamu od začátku (lineární časová složitost $O(N)$).

Oracle nabízí několik typů indexů, jakými jsou například B-stromy (unikátní a s opakováním klíčů, složený index), index s převráceným klíčem, bitmapové indexy. Při



Obrázek 11: Obálka prvku

použití prostorových dat již ale není možné aplikovat indexy navržené pro jednorozměrná data. Nutností se stává použití speciálních postupů indexování prostorových dat, proto je pro nás důležitý prostorový index, který se dá aplikovat na jakýkoliv typ prostorových dat, která máme v databázi uložena.

I prostorových indexů je několik variant, které se liší v principech a algoritmech. Známé prostorové indexy jsou například index dlaždicový, quad-tree (čtyřstrom) index, GiST indexy, Voronoi geodetický index a R-tree (Region-tree, R-strom) index.

Základem každého prostorového indexu je MBR, neboli obálka prvku, která zastupuje minimální a maximální souřadnice obdélníku, který je nejmenším opsaným obdélníkem daného prvku. Zároveň musí být jeho strany rovnoběžné se souřadnicovými osami. Můžeme si jej představit jako obdélník (v některých případech čtverec), který těsně ohraničuje tvar objektu.

5.2.0.0.1 R-tree Za nejefektivnější prostorový index se považuje R-tree, který seskupuje prvky, respektive jejich obálky, podle polohy do hierarchické stromové struktury. Výhodu má oproti quad-tree indexaci fakt, že nevyžaduje tak časté aktualizace. Ve srovnání s dlaždicovým indexem není tak citlivý na velikost prvků díky použití jejich obálek.

Při vykonávání prostorového dotazu při indexaci za pomoci R-tree dojde k průchodu stromu od rootu (kořene stromu) k listům (které mohou obsahovat rozdílný počet záznamů až do předdefinovaného maxima), kde dochází ke kontrole, zda obálka dotazu

překrývá obálku uloženou v aktuálním uzlu. Jestliže ano, prochází dále uvnitř daného uzlu a proces se opakuje.

5.3 Oracle partitioning

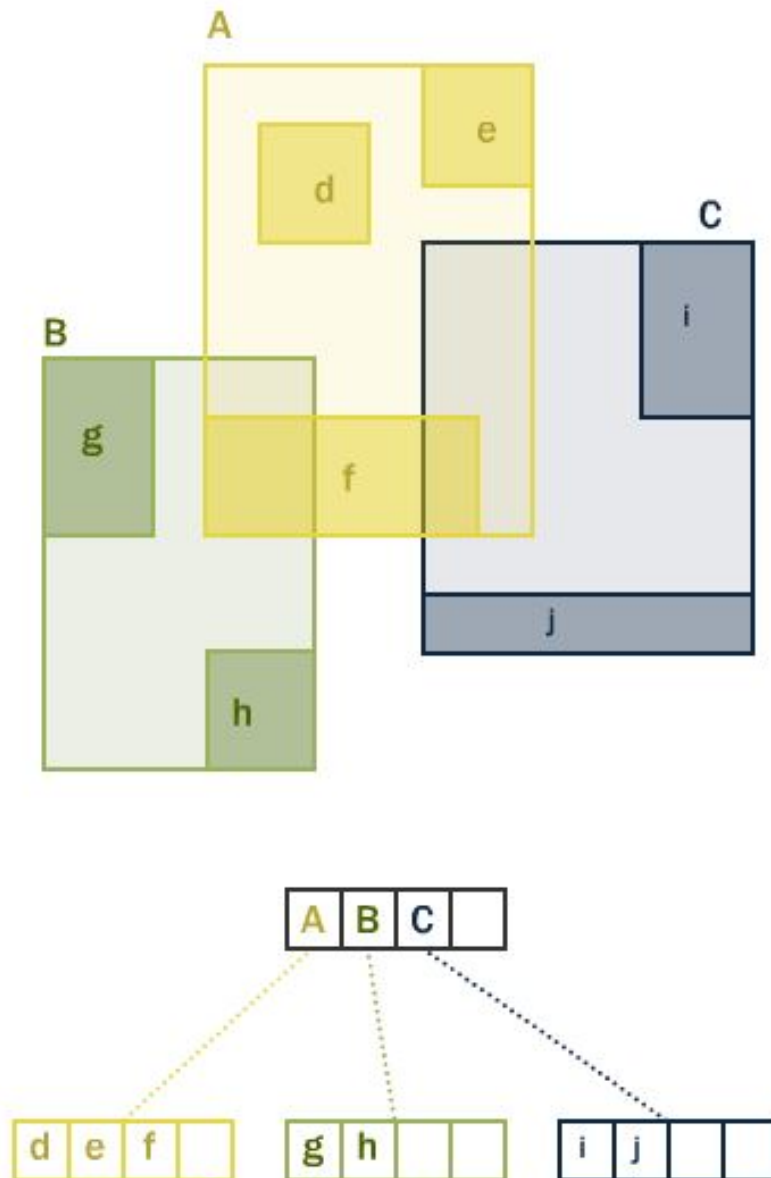
Díky segmentaci je možné zlepšení výkonu, jelikož dochází k rozdělení tabulek a indexů na menší dílky, které se zpracovávají snadněji než jeden větší celek. Jedná se o volitelný doplněk k databázi Oracle, který s sebou přináší zkrácení doby zpracování dotazů, větší dostupnost a lepší zpracovatelnost dat.

5.4 Oracle prostorové dotazy

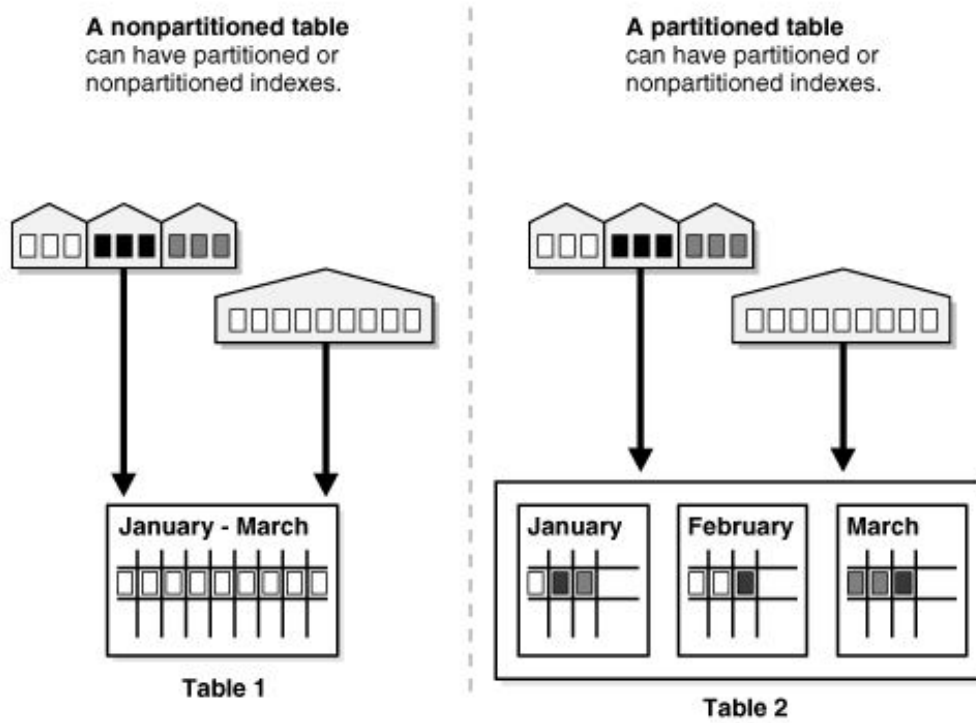
Je speciálním případem dotazu na databázi, který v sobě nese geometrickou složku (SDO_GEOMETRY) zastupující například bod, linii či polygon. Zároveň tyto dotazy berou v úvahu jakýsi vztah mezi zmíněnými geometriemi.

U prostorových dotazů můžeme využít předdefinované funkce, které dokáží pracovat s geometrickými tvary prvků. Dokonce se dá dotazovat i na vícedimenzionální geoprvky. Mezi základní funkce patří zjištění vzájemné vzdálenosti dvou geoprvků, zda jsou si rovny, navzájem se dotýkají, překrývají se, jeden obsahuje jiný, jakou pokrývají plochu, jak jsou dlouhé apod.

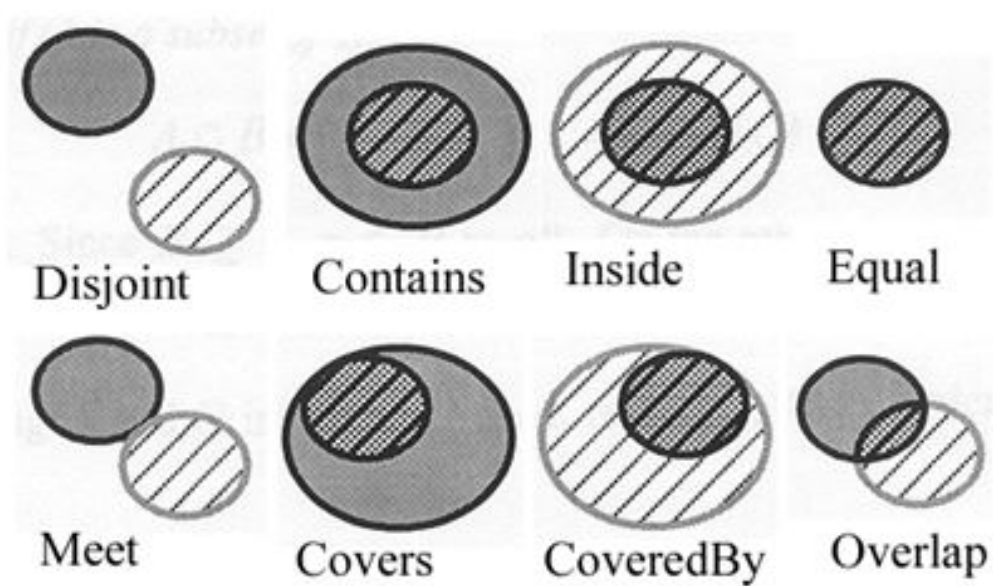
R-tree Hierarchy



Obrázek 12: R-tree



Obrázek 13: Oracle segmentace tabulky



Obrázek 14: Skladba některých binárně-topologických vztahů při prostorovém dotazování

6 Vizualizace prostorových dat

Vizualizací chápeme zobrazování skutečnosti, znázornění dat a informací převedených do grafického zobrazení. S vizualizací se dnes setkáváme na poli mnoha oblastí, jakými jsou strojírenství, architektura, nebo v našem případě geografie. Důvod je jasný – vizualizace slouží k lepšímu pochopení významu dat a vztahů mezi nimi díky své názornosti, která dalece převyšuje vyjádření suchými daty reprezentovanými například tabulkou v elektronickém dokumentu či texty na papíře. Především je obzvlášť těžké vyznat se v obrovském množství dat bez jejich nutné vizualizace.

Nejčastěji je vizualizace prováděna grafy, diagramy, trojrozměrnými objekty nebo skrze mapy, a to za pomoci různých metod. Ty moderní zahrnují počítačové modelování. Vizualizace dat (i prostorových) jako taková má dlouhou historii, kdy v průběhu času samozřejmě prošla vývojem a dnes je běžné používat 3D vizualizace, které jsou však zpravidla náročné na výpočetní výkon a čas. To se dá ovlivnit výkonnějším hardwarem nebo efektivnějším softwarem.

6.1 Metody a algoritmy pro vizualizaci prostorových dat

Aby byla počítačová vizualizace prostorových dat vůbec realizovatelná, je nutné zajistit správný chod programu – software, který má na starosti zpracovávání prostorových dat a jejich reprezentaci, čímž v konečném důsledku vzniká samotná vizualizace těchto dat.

Základním pilířem softwarového vybavení pro vizualizaci prostorových dat jsou, kromě dat samotných, algoritmy, kterými se program řídí a díky nimž je umožněn přístup k datům, jejich zpracování a následný výstup. Postupů a algoritmů pro vizualizaci prostorových dat je několik a zvolení toho optimálního (případně množiny algoritmů) závisí na typu aplikace a požadovaných funkcionalit.

Dnes již není nutné osobně implementovat každý takový algoritmus zvlášť, jelikož existují knihovny a frameworky pro různé programovací jazyky a platformy, které obsahují plno řešení pro zpracování a vizualizaci prostorových dat a práci s nimi zlehčují. Tato řešení jsou postupem času vylepšována a optimalizována zásluhou přispívající komunity. V případě open-source se jedná většinou o komunitu veřejnou, u placených nadstaveb za výsledkem stojí zaměstnanci. Důležité je vědět, které řešení je pro daný účel vhodné, jak funguje a jak s ním pracovat, abychom s úspěchem dosáhli požadovaného výsledku.

6.2 Algoritmy

Jak již bylo zmíněno, algoritmů pro vizualizaci geometrie je spousta a je třeba zvážit výběr a způsob použití spolu s účelem, kterému mají sloužit. V návaznosti na testování zátěže a času pro vykonávání prostorových dotazů, kterým se budu věnovat později, jsem musel některé z těch, které jsem si nastudoval, zavrhnout, jelikož nebyly pro mou desktopovou aplikaci úplně vhodné či dokonce zcela nevhodné.

<http://felinlovewithdata.com/research/the-role-of-algorithms-in-data-visualization>

Algoritmy pro vizualizaci lze rozdělit do několika skupin. První z nich se zabývá způsobem, jakým jsou prostorová data vykreslována a umísťována na obrazovku – toto

je aspekt, který člověk vnímá nejvíce. Do druhé skupiny se dají zařadit algoritmy, které se starají o předzpracování většího množství dat a abstrakci ještě před vizualizací. Do třetí skupiny spadají algoritmy, jejichž úkolem je mapování vlastností dat na vlastnosti vizuální (například jaká data patří do osy X, a která se vážou na osu Y) a optimalizace celého procesu. Poslední skupinu zastupují algoritmy, které zjišťují kvalitu vizualizace.

Zcela odlišným typem algoritmů, který se dá aplikovat na dopravní prostorová data, je algoritmus analýzy těchto dat s následnou predikcí vývoje dopravy pro určitou lokalitu. Tento algoritmus může pracovat na úrovni shody podle známého vzorce vypočítaného na datech z minulosti, nebo je případně založen na principu neuronových sítí.

Mezi první algoritmy, o které jsem se zajímal, se řadil zajímavý algoritmus zvaný squarified treemap, který však, jak jsem později zjistil, nevyužívá pro pozicování souřadnice. V mém případě šlo především o umístění v návaznosti na koordináty.

Vhodnějším kandidátem byl algoritmus pro vizualizaci prostorových dat za pomoci Binned Plot, který přistupuje k jednodimenzionálním datům jako k unikátním přihrádkám, které pojí dohromady s ostatními na základě zeměpisné délky a šířky za použití Mercatorovy projekce. Tento algoritmus se ukázal pro mé účely zbytečně složitý, je vhodný spíše pro sadu obsahující opravdu obrovské množství dat. <http://vis.stanford.edu/files/2013-imMens-EuroVis.pdf>

V mé diplomové práci je využito, mimo jiné, implementace Kd-stromu, jeho vlastností a algoritmů s tím souvisejících, případně kombinací různých algoritmů a návrhových vzorů, které jsou popsány v dalších kapitolách souvisejících s vývojem nástroje pro vizualizaci dopravních dat. Značnou část algoritmů v sobě nesou veřejně dostupné knihovny.

6.3 Kd-tree

K-dimenzionální strom je ve světě prostorových dat uplatnitelný, jelikož dochází k postupnému rozdělování prostoru na podprostor. Vymezením podprostoru lze dosáhnout efektivnějšího hledání. Prostor je rozdělován postupným zpracováním jednotlivých os, což vede k vyváženému stromu (za předpokladu, že bod dělení je medián), jenž v konečném důsledku obsahuje prostory rozdělené až na podprostory, které obsahují už jen jeden jediný bod. Horní výpočetní složitost vytvoření tohoto stromu je $O(n \log^2 n)$ pro n zadaných bodů v případě, že podprostory vznikají půlením mediánem (z toho vyplývá \log^2). U Kd-tree není vždy nutně vyžadováno, aby byl perfektně vyvážený.

6.4 Problém s vizualizací

GIS běžně na obrazovky vykresluje na stovky tisíců (a více) symbolů zastupujících bod, což může být problém z hlediska praktičnosti. Může se stát, že samotný symbol je větší než obrazový bod (pixel), což vede k tomu, že neefektivní vykreslování stále se opakujícího symbolu zabere zbytečně moc času. Symboly se navíc překrývají.

6.4.1 řešení problému

Řešením je eliminace bodů, které jsou mimo rozsah (oblast zájmu), čímž dojde ke zrychlení, jelikož body mohou být organizovány do struktur binárních stromů (KD-tree je sám o sobě speciální případ binárního stromu), které dokáží eliminovat veškeré členy mimo požadovaný rozsah ve výpočetním čase $\log(n)$.

Algoritmy, které se pro počítačovou vizualizaci řadí mezi nejdůležitější, dokáží redukovat složitost vykreslování. Tyto se většinou zaměřují na předcházení vykreslování skrytých ploch v prostoru. Je zbytečné, aby byly vykreslovány například body, které nakonec nejsou vidět či z nich neplyne žádný užitek.

Postupem, který se přímo nabízí, je kontrolovat při každém vykreslení bodu, zda nepřekrývá nějaký bod z množiny bodů již dříve vykreslených. Tedy s každým novým bodem kontrolovat shodu místa všech předchozích, což dává znát, že se jedná o dost náročné řešení (okolo $O(n^2)$, v nejlepším případě $O(n)$ - lineární). Použití této kontroly dává tušit, že se stoupajícím počtem bodů sama o sobě zabere více času než vykreslení všech bodů bez samotné kontroly.

Optimálního vykreslování, které by zobrazovalo pouze body nebo vrcholy, které vizuálně přispívají k rozeznání obsahu na obrazovce, lze dosáhnout i jinými technikami, které závisí na vzdálenosti pozorování. Využívá se zde matematického vztahu mezi odpovídající geografickou vzdáleností pozorovaných bodů a měřítkem, ze kterého se bod stává již neviditelným. Díky tomuto vztahu předem víme, zda bude překrývající bod viděn nebo ne, a tudíž zjistíme, jestli nám tento bod ovlivní zobrazení a chceme jej tedy vykreslit. Tento vztah je následující. Geografická vzdálenost = (Vzdálenost pixelu * rozlišení) / měřítko.

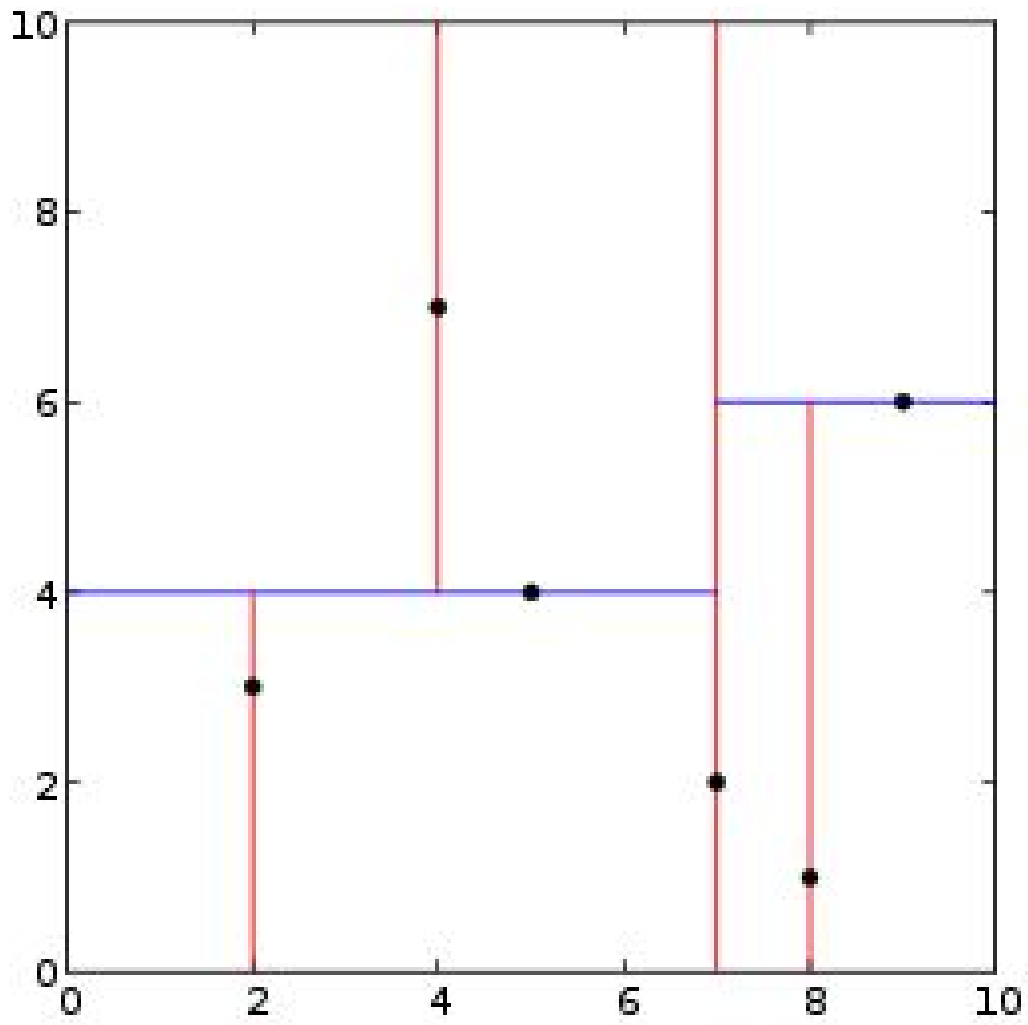
Výhodou tohoto použití je, že velká část nežádoucích bodů je buďto vyřazena výpočtem (při oddálení mapy), nebo z toho důvodu, že se nachází mimo rozsah oblasti zájmu (při přiblížení). Nehledě na úroveň přiblížení se vykreslí jen ta část bodů, která je důležitá.

6.5 Metody pro vykreslování překrývajících se linií

Při vizualizaci překrývajících se linií je nutno vzít v potaz, co mají linie reprezentovat. Řekněme, že v našem případě spojené linie ilustrují silnice. Tyto linie mohou být vykresleny jako přemostění (jedna překrývá druhou) nebo jako křížovatka či souběh (napojují se v jednu). V prvním případě linie vykreslená později překrývá linii starší. V druhém případě dojde ke spojení. Z následujícího obrázku je patrný rozdíl.

6.6 Hermite Interpolation

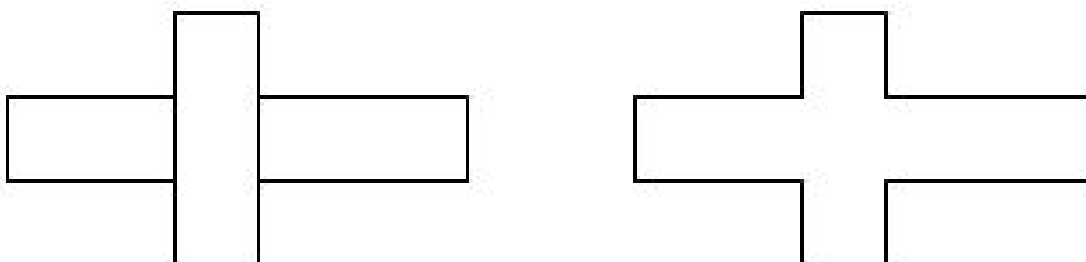
Je metoda interpolace dat (bodů) pomocí polynomiální funkce. Tento algoritmus je možné uplatnit při úpravě grafů, jestliže u vrcholů dojde ke změně z ostrých na plynule zakřivené (jemnější rysy). Úroveň zakřivení okolo bodu udává proměnná bias, tension zastupuje míru napnutí křivky na známých bodech. Pro zachování plynulosti je zapotřebí čtyř bodů (y_0 , y_1 , y_2 a y_3).



Obrázek 15: Kd-tree

```
tuple function build_kd_tree(int depth, set points):  
    if points contains only one point:  
        return that point as a leaf.  
  
    if depth is even:  
        Calculate the median x-value.  
        Create a set of points (pointsLeft) that have x-values less than  
        the median.  
        Create a set of points (pointsRight) that have x-values greater  
        than or equal to the median.  
    else:  
        Calculate the median y-value.  
        Create a set of points (pointsLeft) that have y-values less than  
        the median.  
        Create a set of points (pointsRight) that have y-values greater  
        than or equal to the median.  
  
    treeLeft = build_kd_tree(depth + 1, pointsLeft)  
    treeRight = build_kd_tree(depth + 1, pointsRight)  
  
    return(median, treeLeft, treeRight)
```

Obrázek 16: Pseudokód pro vytvoření dvojdimenzionálního Kd-stromu



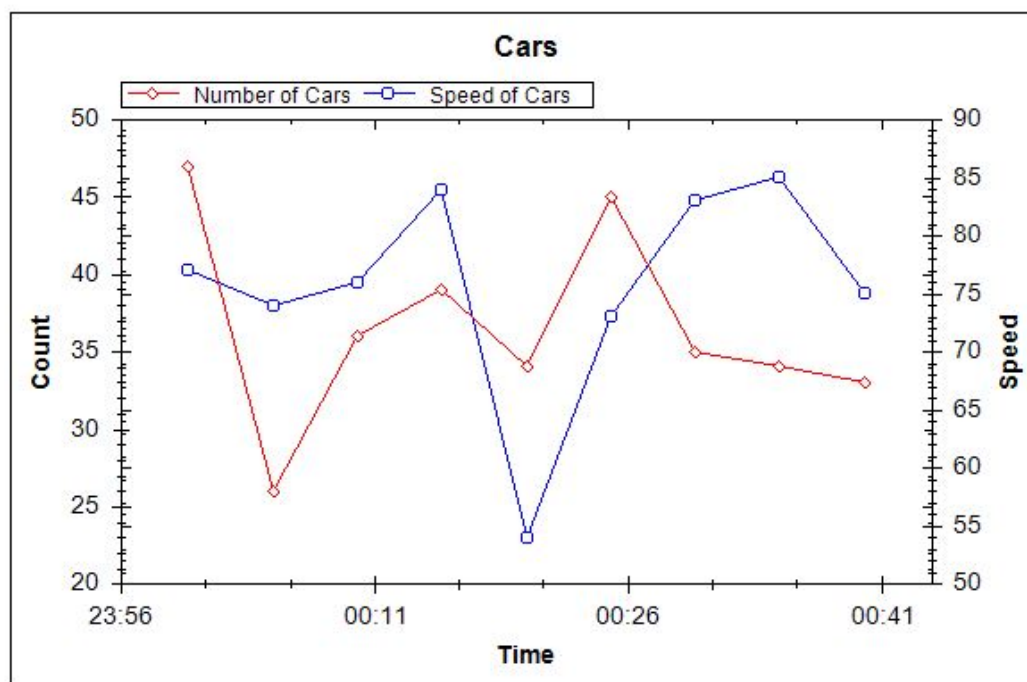
Obrázek 17: Druhy vykreslování překrývajících se linií

```
double HermiteInterpolate(
    double y0, double y1,
    double y2, double y3,
    double mu,
    double tension,
    double bias)
{
    double m0, m1, mu2, mu3;
    double a0, a1, a2, a3;

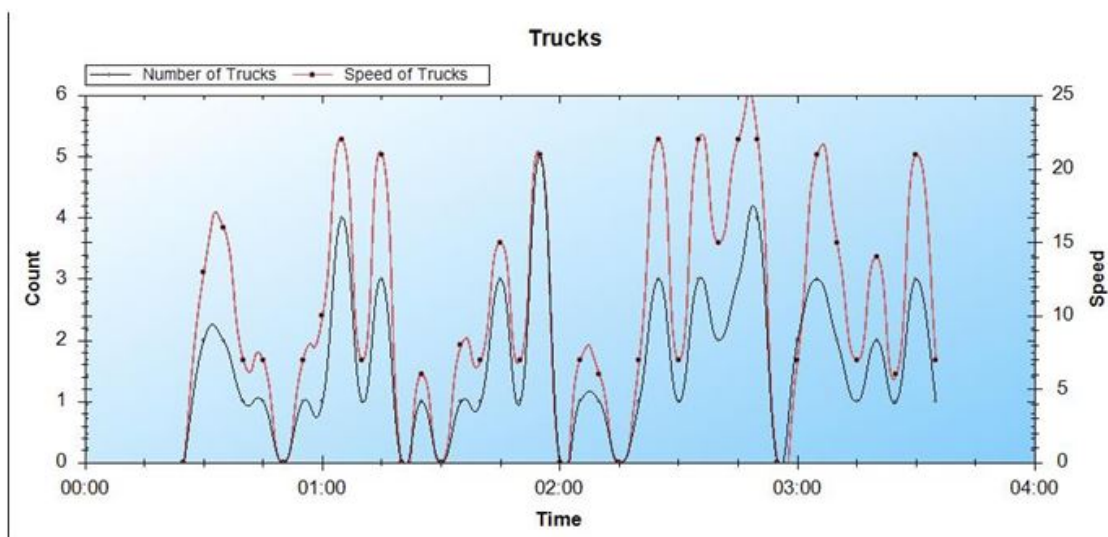
    mu2 = mu * mu;
    mu3 = mu2 * mu;
    m0 = (y1-y0)*(1+bias)*(1-tension)/2;
    m0 += (y2-y1)*(1-bias)*(1-tension)/2;
    m1 = (y2-y1)*(1+bias)*(1-tension)/2;
    m1 += (y3-y2)*(1-bias)*(1-tension)/2;
    a0 = 2*mu3 - 3*mu2 + 1;
    a1 = mu3 - 2*mu2 + mu;
    a2 = mu3 - mu2;
    a3 = -2*mu3 + 3*mu2;

    return (a0*y1+a1*m0+a2*m1+a3*y2);
}
```

Obrázek 18: Funkce na Hermitovskou interpolaci



Obrázek 19: Před použitím Hermitovské interpolace



Obrázek 20: Po použití Hermitovské interpolace

6.7 Frameworky a knihovny

Framework je softwarová struktura vytvořena za účelem podpory při programování a vývoji softwarových projektů. Zlehčuje implementaci díky obsaženým podpůrným programům, knihovnám a návrhovým vzorům. Díky dokumentaci a ukázkám poskytuje i různé rady k implementaci.

Ve své práci jsem používal frameworky Microsoft .NET Framework 4 a DotSpatial ve verzi 1.6, dále knihovnu ZedGraph 5.1.5.

Některé metody a algoritmy, které jsem zmínil, se dají využít díky určitým z dynamických knihoven DotSpatial frameworku, který jsem si pro svou implementaci zvolil v návaznosti na výsledky testů Benchmarku, jenž byl součástí mé diplomové práce. Dalším důležitým aspektem byla jeho vysoká úroveň, obsáhlost složitějších algoritmů (Voronoi Treemaps) a velká pomoc při usnadnění a urychlení práce.

Aktuální zdrojové kódy tohoto frameworku jsou dostupné na <https://dotspatial.codeplex.com/SourceCode>

Framework .NET jsem si zvolil, jelikož jsem díky němu po dobu studia naimplementoval na desítky desktopových i webových aplikací a informačních systémů. Za ten čas jsem se v něm naučil alespoň zčásti orientovat a zjistil jsem, že spolu s jazykem C# a Oracle PLSQL tvoří kombinaci, která mi vyhovuje nejvíce.

ZedGraph je knihovna tříd rozhraní .NET napsána v jazyce C#, která slouží k vizualizaci grafů na základě dat (například věk), dodatečných informací (například, že věk je uváděn v letech) a nastavení (například sloupcový diagram). V tomto případě sloupcový graf může vyjadřovat průměrný věk obyvatel podle státu, kdy osa X zastupuje název státu a osa Y vyjadřuje věk v letech.

Původně jsem zvažoval i využití projektu OpenStreetMap a JavaScriptového API WebGL, případně i Gogole Earth za pomoci KML souboru založeném na XML. Cílem práce však bylo vytvořit standalone desktopovou aplikaci, tedy nástroj pracující samostatně i offline (samozřejmě v omezeném režimu bez načítání online informací), bez nutnosti předávat data ke zpracování na Internet, následkem čehož by došlo ke zveřejnění.

7 Postup prací

V této kapitole jsou uvedeny postupy od zadání práce až k jejímu dokončení. Během celkového procesu jsem se potýkal s instalací a nastavením Oracle serveru, připojení a účtů. Pokračoval jsem vytvářením skriptů a procedur na straně serveru, vytvářením a optimalizací tabulek a dat spolu s indexací a segmentací tabulek prostorových dat, nad kterými jsem dále prováděl výkonnostní testy. Vyzkoušel jsem si načítání souřadnic do Google Maps z vlastní databáze a v poslední fázi implementaci podpůrných konzolových aplikací pro vlastní testovací aplikaci spolu s desktopovým nástrojem pro vizualizaci dopravních dat.

Skripty, tabulky, textové logy, grafy, obrázky a zdrojové kódy jsou uvedeny v neveřejné části v přílohách. Zde budou uvedeny jen některé části.

7.1 Oracle Server a SQL Developer

Instalace Oracle Database 12c je poměrně snadnou a rychlou záležitostí, takže by měla probíhat bez větších problémů. Dá se říci, že stačí vyplnit pár kolonek, vybrat z několika nabídek, zapamatovat si důležitá hesla a umístění a dokončit průvodce. Tato verze pro jeden server je optimalizována pro vývoj v malých podnicích a pro účely této diplomové práce byla naprosto dostačující.

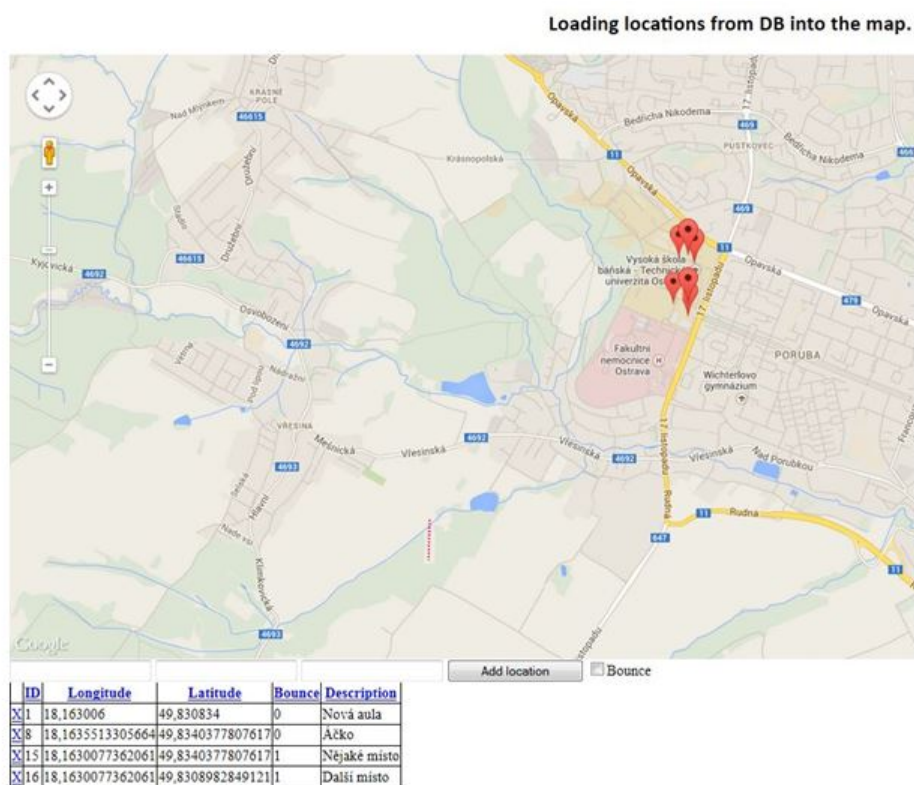
Po úspěšné instalaci jsem nenašel důvod vracet se do grafického rozhraní Database Configuration Assistant, které Oracle nabízí. Veškeré úkony, které jsem potřeboval provést, se daly zprostředkovat příkazy v Oracle SQL Developer, což je grafické uživatelské prostředí umožňující práci s Oracle databází. Jakmile jsem si vytvořil v databázi účet a byl jsem schopen se na něj připojit přes SQL Developer, mohla začít práce s daty a první pokusy s vizualizací. Pokud je vyžadovaná práce s geodatami, je dobré nainstalovat rozšíření GeoRaptor, který je schopen zobrazovat prostorové záznamy z databáze a do jisté míry s nimi pracovat. Tento doplněk jsem používal dokud mnou vyvíjená aplikace nebyla sama schopna obstarat zobrazení prostorových záznamů.

7.2 Použití Google Maps

Prvním krokem byl zkušební projekt napsaný v C# a JavaScriptu za pomoci AJAXu a ASP.NET technologie. Pro jednoduchou tabulku vystačil SQL Server Database. Tabulka obsahuje pár záznamů, které uchovávají informace o bodu, jakými jsou jeho název, zeměpisná délka a šířka.

Jelikož je databáze a tabulka pouze lokální, není pro načtení ani vložení záznamů zapotřebí připojení k Internetu. Pro vykreslení bodů do mapy už však ano. Online se tato data při spuštění načtou a v internetovém prohlížeči se díky souřadnicím (v WGS84) vykreslí do Google Maps.

Přestože toto řešení nebylo v žádném případě vhodné a nemínil jsem v něm pokračovat, naučil jsem se základní práci s latitudou a longitudou.



Obrázek 21: Vykreslení bodů z databáze do Google Maps

7.3 Benchmark

Aby byly zvoleny vhodné algoritmy, které je možné při implementaci vizualizace prostorových dat použít, je nutné znát rychlost načtení těchto dat ze serveru, jakmile si o ně koncový uživatel požádá. Čas získání výsledku může ovlivnit několik faktorů, jakými jsou rychlost připojení, rychlost vykonání dotazu nebo uložené procedury či propustnost databáze. Velkou roli může hrát i propustnost VPN, pokud nejsme přímo připojeni k síti, ve které se server nachází a je nutné použít tunelového propojení. Toto vše je třeba zohlednit.

K tomuto účelu jsem implementoval benchmark – výkonnostní test, který má za úkol ověřit zlepšení při určitých změnách, které mají vést k optimalizaci. Jestliže je čas mezi požadavkem, zpracováním a poskytnutím dat příliš vysoký, není vhodné použít algoritmy s větší časovou náročností. Optimalizace může být realizována na straně serveru (například indexací dat), upravením dotazů a procedur, nebo na straně aplikace, kde je možné zachránit větší časovou prodlevu rychlejším zpracováním dat.

Benchmark je implementován v C# .NET Framework 4. Pro vizualizaci grafu byla použita knihovna MSChart.

7.3.1 a podmínky potřebné pro testování

Funkční nainstalovaný Oracle server, 16 založených uživatelů s povoleným připojením a právy načítání dat z tabulek, které obsahují přes sto milionů skriptem nahodile vygenerovaných prostorových dat (bodů) na území České republiky a okresy České Republiky zastoupeny polygony. Musí být zajištěno internetové připojení.

7.3.2 Testovací uživatelé a tabulky

Nejdříve je nutné založit 16 účtů, které poslouží k účelům testování. To se dá snadno provést za pomoci několika málo příkazů nutných pro správný chod.

```
CREATE USER pc1 IDENTIFIED BY databaze;
```

```
GRANT CONNECT TO pc1; --povoleni pripojeni
GRANT SELECT on okresy TO pc1; --povoleni selectu na tabulku okresy
GRANT SELECT on body TO pc1; --povoleni selectu na tabulku bodu
```

Je také dobré si udělat zálohu tabulek a případně na těchto tabulkách zkusit různé indexování nebo segmentaci tabulky pro porovnání výsledků s původní tabulkou.

```
GRANT SELECT on body_partitioned TO pc1; --povoleni selectu na segmentovanou tabulku
```

Původní tabulka Body obsahovala souřadnice v datovém typu SDO_GEOMETRY. Tyto souřadnice můžeme získat například následujícím dotazem.

```
SELECT c.point_id, t.X, t.Y FROM body c, TABLE(SDO_UTIL.GETVERTICES(c.geo)) t;
```

Vytvořil jsem ještě jednu kopii této tabulky, ve které jsem skriptem docílil update, který z SDO_GEOMETRY vytáhl souřadnice a uložil je do nových sloupců X a Y pod datovým typem number. Toto jsem provedl pro všechny záznamy, abych mohl porovnat,

```
alter table body_copy
ADD
(
    X          number,
    Y          number
);

UPDATE body_copy t1
SET (X, Y) = (SELECT t2.X, t2.Y
              FROM body t2
              WHERE t1.point_id = t2.point_id)
WHERE EXISTS (
    SELECT 1
    FROM body t2
    WHERE t1.point_id = t2.point_id )
```

Obrázek 22: Alter table a update

zda se spolu s datovými typy při dotazování bude lišit i čas vykonání. Přidání nových atributů a update nad tabulkou Body_copy může vypadat takto.

8 Testování

Testovací aplikace testuje data ve čtyřech různých typech testů, které závisí na následujících dotazech a módech spuštění. V případě, že je dotaz složený z proměnných `right` a `left`, pak se za tyto proměnné dosazují hraniční hodnoty vypočtené v závislosti na počtu testovaných dotazů spuštěných najednou. Testování vždy probíhá paralelně v počtu 16 různých uživatelů, což simuluje případ, kdy se o určitý počet různých dat uchází více uživatelů najednou ve stejný čas. Výsledek dává znát, jak server tyto požadavky zpracovává a pomáhá tak odhalit slabší místa.

Příklad: Jestliže nastavíme množství dotazů na 160, rozdělí se tento počet mezi 16 uživatelů rovnoměrně (na každého tedy připadá dotazování se na 10 různých prostorových dat), propočte se unikátní rozmezí řádků z tabulky bodů obsahující prostorová data, na které se uživatel dotazuje. Pro každého uživatele se v tomto případě určí 10 unikátních záznamů.

Tedy uživatel č.1 bude testovat z rozmezí prvního záznamu (`left=0`) až desátého záznamu (`right=9`), uživatel č.2 pokračuje hranicí `left=10` až `right=19`, až dojde k celkovému rozdělení.

Testovací konzoli je možno spustit pouze pro testování, nebo případně i zobrazení, který uživatel je na serveru aktivní (zda opravdu dotazování probíhá paralelně) a případně, zda u něj stále testování probíhá, nebo již bylo dokončeno. Dalším výstupem je doba, za kterou došlo u všech 16 uživatelů k připojení k databázi (tedy byli připojeni všichni najednou). Hlavním výstupem však zůstává seznam všech uživatelů a časů, za něž obdrželi dotazovaná data.

8.1 Typy testů

Nejdříve jsem si pro měření času napsal jednoduchou uloženou proceduru v `plsql`. Ukázalo se však, že měření času nebylo úplně adekvátní. Rozhodl jsem se tedy využít přímo dotazy volané z testovací aplikace.

První typ testu získává prostorová data datového typu `sdo_geometry` a každé ze získaných načtených dat tohoto typu reprezentující bod na území České Republiky se zařadí do okresu na území České Republiky (taktéž reprezentovaný datovým typem `sdo_geometry`), do kterého patří.

Druhý typ testu zjišťuje, kolik času zabere načtení určitého bodu na území ČR reprezentovaného datovým typem `sdo_geometry` bez zařazení do okresu na území ČR.

Třetí typ testu se zaměřuje podobně jako první typ na zařazení bodu do příslušného okresu na území ČR s tím rozdílem, že body na území ČR nejsou prezentovány datovým typem `sdo_geometry`, ale souřadnicemi `X` a `Y` v zastoupení datového typu `number`.

Čtvrtý typ načítá souřadnice `X` a `Y`, tedy podobně jako druhý typ testu, akorát se záměnou datového typu `sdo_geometry` za datový typ `number`.

Defaultně je v testovací aplikaci spuštěn třetí typ testu, jelikož nejvíce reflektuje použití dotazování v nástroji pro vizualizaci prostorových dat.

Dotaz 1:

```
select Point_id, Nazev from ( select rownum rnum,
a.point_id, a.nazev
from
(SELECT p. point_id, o.nazev,      rownum r FROM
points2d p, okresy o
WHERE
SDO_CONTAINS(o.geom, p.geom )= 'TRUE') a
WHERE
rownum <= " + right + " )where rnum >= " + left;
```

Obrázek 23: První typ testu

Dotaz 2:

```
select geo
from
(select rownum rnum, a.* from (SELECT p. geo, rownum r
FROM points2d p) a
WHERE
rownum <= " + right + " ) where rnum >= " + left;
```

Obrázek 24: Druhý typ testu

Dotaz 3:

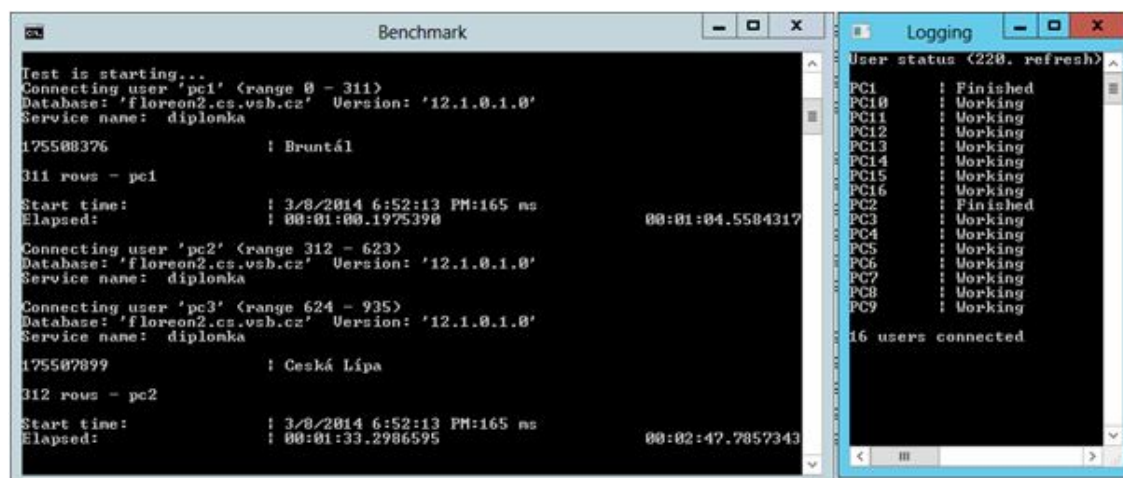
```
select Point_id, Nazev from ( select rownum rnum,
a.point_id, a.nazev
from
(SELECT p. point_id, o.nazev, rownum r FROM points2d p,
okresy o
WHERE
SDO_CONTAINS(o.geom,
MDSYS.SDO_GEOMETRY(2001,NULL,MDSYS.SDO_POINT_TYPE(p.X,p.Y,N
ULL),NULL,NULL))='TRUE') a
WHERE
rownum <= " + right + " )where rnum >= " + left;
```

Obrázek 25: Třetí typ testu

Dotaz 4:

```
select X, Y from points2d
where rownum <= " + right + " and rownum >= " + left;
```

Obrázek 26: Čtvrtý typ testu



Obrázek 27: Test spuštěn z příkazové řádky nebo nástroje pro vizualizaci dopravních dat – průběh

8.2 Poznámka k vydání aplikace a případná zjištění

Postupným vývojem testovací aplikace vzniklo několik verzí. Poslední Benchmark verze zvládá spuštění testování v závislosti na nastavení v kódu nebo z příkazové řádky pomocí argumentů, čehož se využívá v nástroji pro vizualizaci prostorových dat, jehož je Benchmark dílčí součástí.

Příklad: TestConsole.exe 160 yes yes – 160 = počet dotazů – yes = přehled stavu uživatelů zapnut – yes = po skončení vypsání délky připojení V případě přehledu stavu uživatelů se spustí druhá konzole, která v pravidelných intervalech kontroluje a vypisuje stav dění uživatelů během testování. Jestliže navíc chceme znát i délku připojení uživatelů, spustí se podproces, ze kterého si druhá konzole vypíše data. Tyto možnosti lze samozřejmě kombinovat.

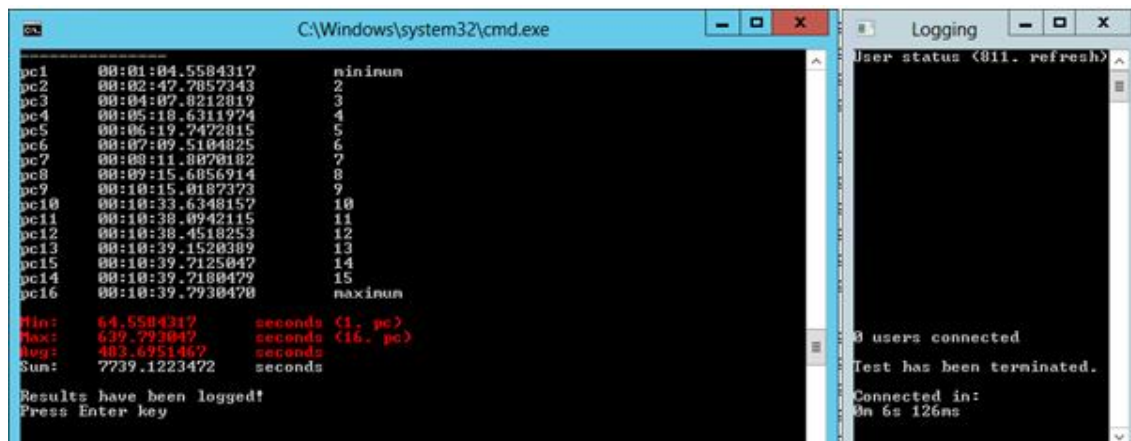
Příklad 2: Testconsole.exe 160 – spustí pouze test

Jakmile je zadán počet simultánních dotazů, propočítají se hranice a přidělí se každému uživateli, jak bylo popsáno výše. Zde začíná opravdové testování, ať už I. relačně-prostorové v rámci jednoho dotazu II. v rámci měření rozděleného do dvou fází (příprava bodů z rozmezí – relační část a postupné zařazení těchto bodů do krajů – prostorová část dotazu).

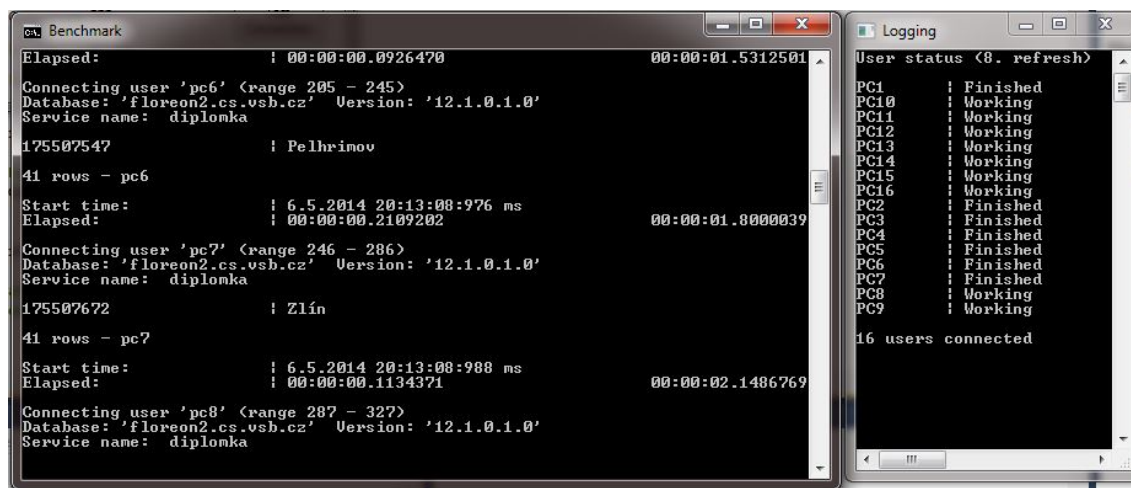
Tyto časy se dají porovnat. Čas získání pouze prostorových dat (II. typ testu) je velmi podobný času získání okresu – zařazení bodu na základě získaných prostorových dat (I. typ testu) při nízkém počtu dotazů. V zásadě se délka dotazování a získání dat příliš neliší. V případě dotazování se na zařazování při větším počtu bodů čas vzrůstá.

U dotazování se na body čas vzrůstá až v řádech milionu bodů, čehož se podařilo dosáhnout využitím Oracle partitioningu.

Dalo by se tedy konstatovat, že vypočítané časy získané testováním jsou v obou případech podobné, případně je efektivnější se dotazovat relačně a prostorově dohro-



Obrázek 28: Test spuštěn z příkazové řádky nebo nástroje pro vizualizaci dopravních dat – konec



Obrázek 29: Benchmark.exe – výpočet rozmezí a start testu


```
result = test.testNow(typeOfTest, testRowNum);
        CreateLog(result, testRowNum.ToString(), typeOfTest);
```

Obrázek 30: Zahájení testu a uložení vygenerovaných souborů

mady. Většinu času totiž nezabere relační dotaz, kdy se snažíme dostat prostorová data bodu, ale pozdější zařazení těchto bodů do patřičných regionů, do kterých body patří na základě prostorových dat. Toto se povedlo urychlit v případě užití vyextrahovaných souřadnic X a Y namísto použití `sdo.geometry`.

Je tedy efektivní provést operace nejlépe najednou (atomicky, jako jednu transakci), kdy dojde k vyhledání polohy bodu a následnému vyhledání, ve které oblasti se bod se získanými souřadnicemi nachází – zařazení bodů do souvisejících okresů, ve kterých leží (operator `SDO_CONTAINS`).

Jakmile test skončí, vygeneruje se automaticky textový záznam výsledků a znázornění těchto dat grafem. Oba soubory se uloží na pevný disk.

8.3 Assembly dependencies

Aplikace Benchmark se skládá z několika konzolových aplikací. Vztahy mezi nimi je patrná na následujícím obrázku.

8.4 Výsledky testů

V této kapitole uvádím výběr z naměřených hodnot v tabulkách, grafech a textovém výstupu.

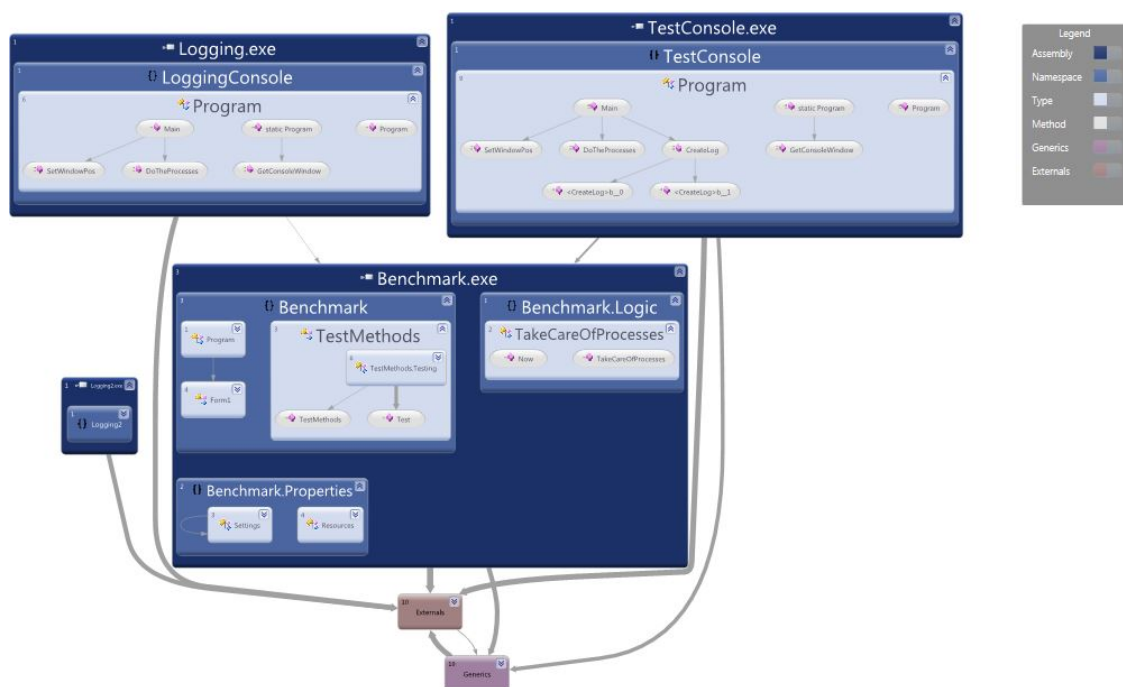
Oracle nabízí v základu využití pouze jednoho jádra. Běh na více jádrech se platí podle licence na každé další jádro. Všechny testy, které jsem spustil, využívaly pouze jedno jádro, což jistě značně ovlivnilo jejich výsledky, stejně tak technické parametry a konfigurace serveru, který Benchmark využíval.

Oracle Database běží na Windows Server 2012 Standard 64-bit s dvoujádrovým procesorem X5355 taktovaným na 2.66 GHz a RAM 8 GB.

8.5 •

Tabulky Následující tabulky zastupují naměřené hodnoty v závislosti na počtu dotazů před přidáním prostorového indexu založeném na R-tree a po jeho přidání.

Testy staršího data (ze dne 21. a 25.1.2014) byly spuštěny v samostatné testovací aplikaci. Testy data novějšího (30.4.2014) již byly spuštěny jako součást nástroje pro vizualizaci dat.



Obrázek 31: Assembly dependencies aplikace Benchmark

8.6 Bez prostorového indexu

8.6.1 S prostorovým indexem

Následující tabulka udává nejlepší naměřené časy, kterých jsem za celou dobu testování dosáhl. Indexování tedy přispělo k optimalizaci, přestože délka trvání stále optimální není.

8.7 Grafy

Po ukončení každého textu se vygeneruje sloupcový graf, který vyjadřuje na ose X konkrétní uživatele seřazené podle jména a jejich přesné časy. Na ose Y se nachází automaticky dopočítaná časová osa uváděná v sekundách sloužící jako škála.

Každý graf má v hlavičce uvedeno, o jaký typ testu se jedná, a počet zařazení bodu do okruhu, kterému bod náleží.

Při bližším porovnání grafů s vyšším počtem dotazů před a po indexaci je patrný rozdílný nárůst změřených časů. Po indexaci narůstá déle a pozvolně. U neindexovaných dat sloupce stoupají dřív a téměř s pravidelností.

Podle vizualizace můžeme říct, že Oracle server zpracoval naše dotazy v dávkách. Prvních několik je zpracováno téměř ihned. Druhou skupinu zpracovává postupně a ve třetí vlně mají uživatelé opět velmi podobné časy (to příkládám faktu, že jich pro obsluhu už zbývá jen málo).

⊕ Druhý typ testu bez partitioningu. (24.1.14-25.1.14)

Počet dotazů	Název PC / nejkratší čas (m:s:ms)	Název PC / nejdelší čas (m:s:ms)
16	14 / 00.7048808	1 / 04.7937773
1600	15 / 00.7121198	8 / 03.2209522
16000	6 / 01.0256294	8 / 03.1010533
15000000	3 / 10:43.5272691	7 / 15:06.2282469

Třetí typ testu bez partitioningu. (24.1.14)

Počet dotazů	Název PC / nejkratší čas (m:s:ms)	Název PC / nejdelší čas (m:s:ms)
16	12 / 00.4627492	2 / 05.0251230
1600	3 / 02.2502802	10 / 02:01.6858817
16000	1 / 03:13.6426029	16 / 39:11.9077795

Druhý typ testu s partitioningem. (25.1.14)

Počet dotazů	Název PC / nejkratší čas (m:s:ms)	Název PC / nejdelší čas (m:s:ms)
16	12 / 00.4627492	2 / 05.0251230
1600	6 / 00.6732315	1 / 04.4953883
16000	12 / 00.9895085	7 / 04.4996983
15000000	10 / 02.9018323	1 / 01:44.6538305

Třetí typ testu s partitioningem bodů. (29.1.14)

Počet dotazů	Název PC / nejkratší čas (m:s:ms)	Název PC / nejdelší čas (m:s:ms)
16	2 / 00.7966764	1 / 07.0533929
1600	2 / 01.8057414	11 / 02:30.2891656
16000	NA	NA

Obrázek 32

☒ Třetí typ testu bez partitioningu. (30.4.14)

Počet dotazů	Název PC / nejkratší čas (m:s:ms)	Název PC / nejdelší čas (m:s:ms)
16	4 / 00:00.1817615	3 / 00:00.5311207
32	5 / 00:00.1472289	1 / 00:00.4622145
64	4 / 00:00.1764591	3 / 00:00.5636047
128	8 / 00:00.4579103	1 / 00:01.0575964
256	4 / 00:00.3609153	14 / 00:01.2907494
512	1 / 00:00.4695953	14 / 00:02.5821530
1024	1 / 00:00.7682133	11 / 01:29.2402080
1600	1 / 00:00.9135039	12 / 03:11.5467525

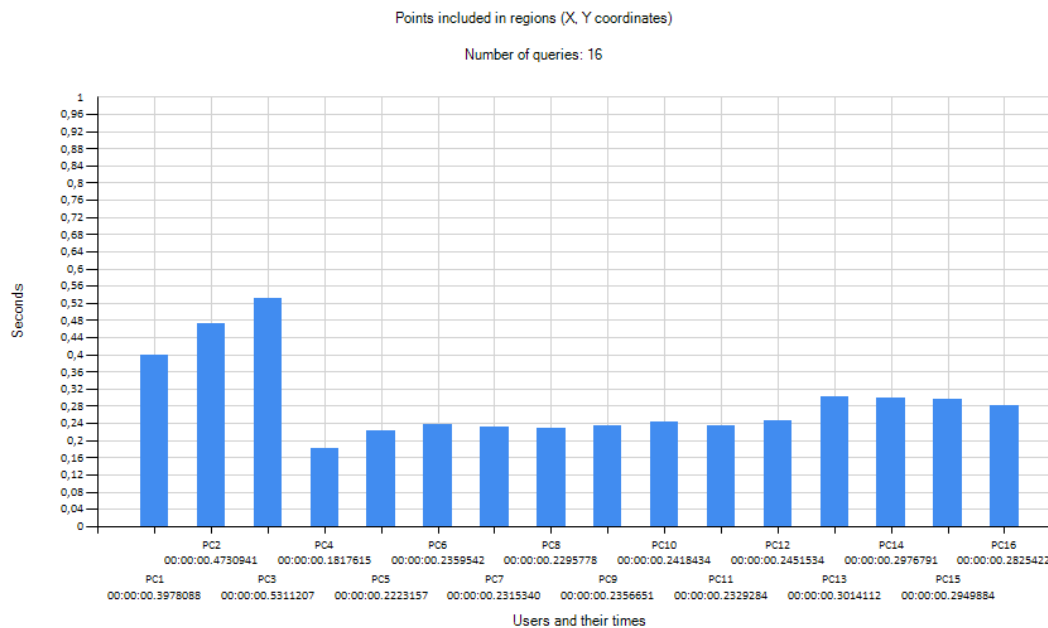
- NA – testování nebylo spuštěno, jelikož na serveru běžela časově náročná procedura
 – ze stejného důvodu mohly být naměřené časy o něco větší

Obrázek 33

Třetí typ testu bez partitioningu. (30.4.14)

Počet dotazů	Název PC / nejkratší čas (m:s:ms)	Název PC / nejdelší čas (m:s:ms)
16	4 / 00:00.1300577	2 / 00:00.5227899
32	1 / 00:00.0274743	14 / 00:00.1268797
64	1 / 00:00.0178450	14 / 00:00.2532602
128	4 / 00:00.2135730	15 / 00:00.7231858
256	1 / 00:00.0758176	15 / 00:01.2366003
512	4 / 00:00.4200941	15 / 00:01.5087873
1024	1 / 00:00.7352836	10 / 00:38.5351228
1600	1 / 00:00.7821642	12 / 00:57.2066344

Obrázek 34



Obrázek 35: Graf třetího testu bez indexu, počet dotazů 16

8.7.1 Bez prostorového indexu

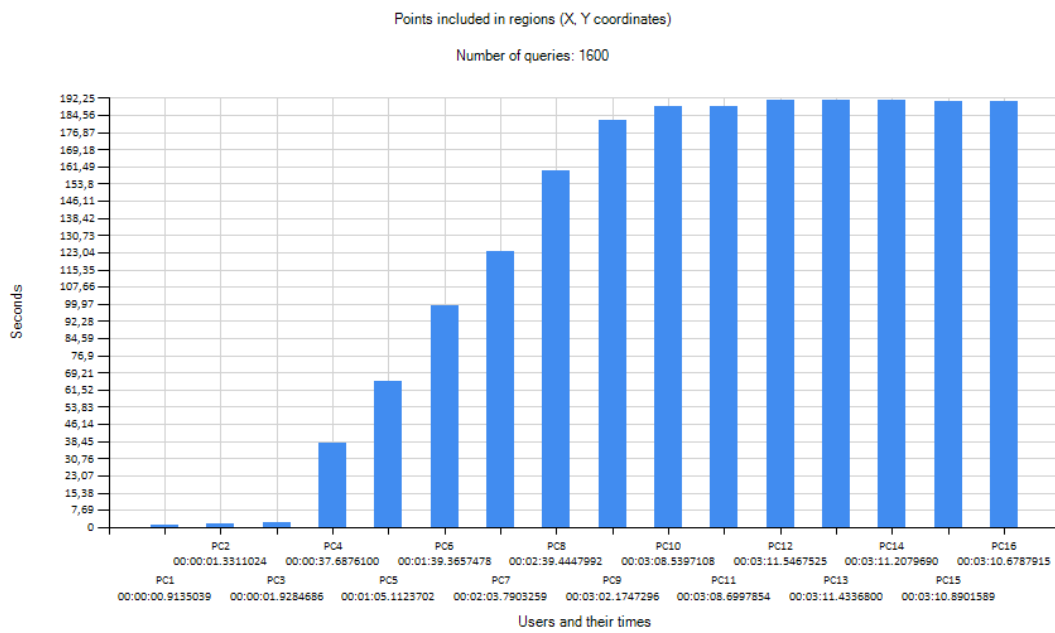
Následující grafy reprezentují první a poslední záznam třetího typu testu bez partitioningu bez indexace ze dne 30.4.2014, který je tabulkou vyjádřen výše.

8.7.2 S prostorovým indexem

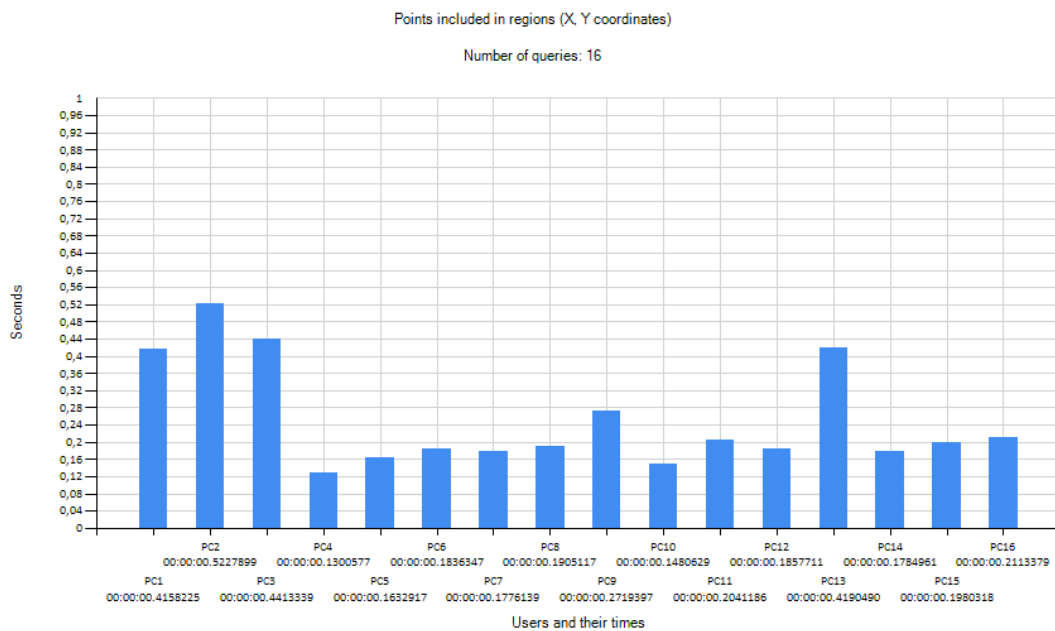
Následující grafy reprezentují první a poslední záznam třetího typu testu bez partitioningu s indexací ze dne 30.4.2014, který je tabulkou vyjádřen výše.

8.8 Textový výpis

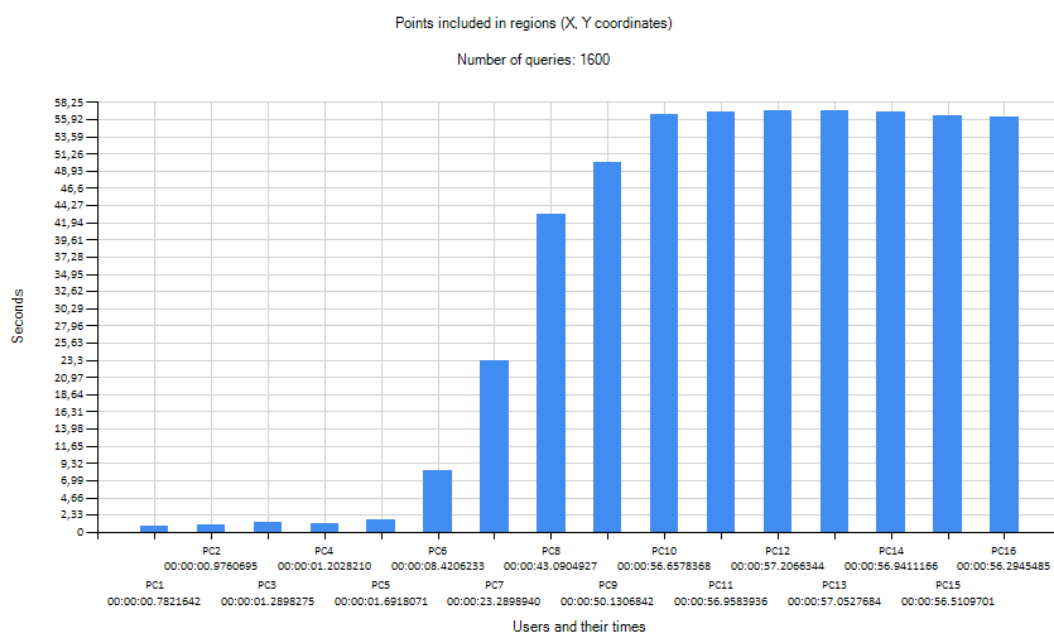
Textový výpis je záznam výsledků testů v podobě prostého textu. Není tak přehledný jako grafy, ale nechávám si jej čistě z důvodu logování výsledků.



Obrázek 36: Graf třetího testu bez indexu, počet dotazů 1600



Obrázek 37: Graf třetího testu s indexem, počet dotazů 16



Obrázek 38: Graf třetího testu s indexem, počet dotazů 1600

Points included in regions (X, Y coordinates)
Number of queries: 1600

User	Time
1	00:00:00.7821642
2	00:00:00.9760695
3	00:00:01.2898275
4	00:00:01.2028210
5	00:00:01.6918071
6	00:00:08.4206233
7	00:00:23.2898940
8	00:00:43.0904927
9	00:00:50.1306842
10	00:00:56.6578368
11	00:00:56.9583936
12	00:00:57.2066344
13	00:00:57.0527684
14	00:00:56.9411166
15	00:00:56.5109701
16	00:00:56.2945485

ordered by time (ascending)

1	00:00:00.7821642
2	00:00:00.9760695
4	00:00:01.2028210
3	00:00:01.2898275
5	00:00:01.6918071
6	00:00:08.4206233
7	00:00:23.2898940
8	00:00:43.0904927
9	00:00:50.1306842
16	00:00:56.2945485
15	00:00:56.5109701
10	00:00:56.6578368
14	00:00:56.9411166
11	00:00:56.9583936
13	00:00:57.0527684
12	00:00:57.2066344

Obrázek 39: Textový výpis výsledku třetího testu s indexem, počet dotazů 1600

9 Nástroj pro vizualizaci dopravních dat

Tato kapitola popisuje vývoj a využití aplikace pro vizualizaci dopravních dat s použitím prostorové databáze Oracle.

Celý nástroj jsem implementoval ve vývojovém prostředí Visual Studio 2010 za použití jazyka C# pod .NET a DotSpatial frameworkem s využitím ZedGraph knihovny a Oracle Spatial modulu. Prostorová data byla uložena v Oracle databázi běžící na Oracle Server 12c. Data, která nebylo nutné načítat opakovaně ze serveru, jelikož se nemění (například shapefile s oblastí České Republiky), jsem automaticky přidával z uloženého souboru, který jsem stáhl z veřejně dostupných zdrojů nebo jej vygeneroval z prostorových dat uložených v databázi.

Veškeré zdrojové kódy, skripty a data jsou součástí neveřejné práce a zde uvedu jen vybrané ukázky.

9.1 Co je naimplementováno

Aplikace si rozvrhne vykreslení komponent v závislosti na rozlišení obrazovky. Po otevření aplikace se automaticky načtou hned dvě vrstvy. Obrys České Republiky a body znázorňující mýtné brány na území České Republiky. V tuto chvíli je aplikace i mapa připravená a je do ní možné přidávat další vrstvy nebo je naopak z ní odebírat či schovávat. Některé prvky vrstvy se dají modifikovat. Například název, barva a symbol objektů ve vrstvě. Je také možné načíst atributy zobrazovaného prvku, pokud jej obsahuje (například název města).

Mapu je možné přibližovat a oddalovat, což má za následek přepočítání zobrazovaných bodů. Je možné s ní pohybovat nebo ji úplně smazat. Při každém pohybu myši nad mapou se ve stavovém řádku zobrazují souřadnice místa, nad kterým se ukazatel myši pohybuje. Čím větší přiblížení mapy, tím přesnější souřadnice lze dostat. Souřadnice by měly být ve stejném souřadnicovém systému. Pokud aplikace rozezná rozdílný souřadnicový systém, nabídne uživateli reprojekci.

Zvolené souřadnice lze přidávat do seznamu míst. Vybrané pozice nemusí nutně odpovídat přesné poloze mýtné brány, proto se vždy nejdříve propočítá ta nejbližší zvolené lokalitě.

V případě zvolené mýtné brány a data se vyhledá maximálně tolik prvních záznamů souvisejících s pozicí a datem, kolik si uživatel nastaví. Po načtení a zobrazení dat lze vybrat pouze určité záznamy. Jakmile jsou záznamy vybrány, je nutné zvolit, zda se mají načíst hodnoty pro auta, nebo nákladní vozy.

Po vybrání jsou k dispozici údaje o maximální rychlosti a počtu aut či nákladních vozů ve zvoleném časovém horizontu na vybraném úseku dálnice. K dispozici jsou dva typy grafů pro vykreslení načtených dat. První z nich vykreslí graf okamžitě, druhý postupně do grafu přidává data, čímž dochází k primitivní animaci. Grafy jsou interaktivní, takže se vlivem najetí myši na body reprezentující počet a rychlost vozidel v čase zobrazí přesné hodnoty. U grafů je možnost přiblížení, oddálení a uložení grafu jako obrázek.

Cars	Highway: D5
Maximal Count: 32	Direction: Rozvadov
Maximal Speed: 102	Kilometrage: 137,13km

Obrázek 40: Výpis základních informací o zvoleném segmentu

9.2 Načítání dat do mapy

Mapa je komponenta, která dokáže zobrazit vektorová a rastrová data pod zvoleným systémem souřadnic, který je ve výchozím nastavení WGS84. Pokud je potřeba jiného souřadnicového systému, je potřeba provést transformaci.

Přidání lze uskutečnit přes OpenFileDialog, prostým přetažením souboru na plochu mapy nebo načtením z databáze. Veškerá přidaná prostorová data jsou zobrazena v legendě jako vrstvy a vykreslena do mapy (pokud není vrstva vypnutá).

Pro přidávání vrstev do mapy se využívá algoritmu Kd-tree, který byl v práci popsán.

9.3 Určení Latitude a Longitude na mapě

Souřadnice bodů se propočítávají v závislosti na pozici myši. Zde je krátký kód znázorňující jak přepočtu docílit při každém pohybu myši nad mapou. Řešení není složité díky použití DotSpatial frameworku.

Uložení koordinátů vybraného bodu může vypadat nějak takto. Osobně tento kus kódu využívám v události pro dvojklik myši.

Není potřeba volat metodu PixelToProj(Positron positron) znovu, jelikož ta se volá s každým pohybem ukazatele myši nad mapou – a v jiném případě nemá ukládání pozice smysl.

9.4 Využití grafů

Na následujícím grafu je vidět patrná korelace mezi rychlostí a počtem nákladních vozů na určitém segmentu mezi půlnocí a čtvrtou hodinou ranní. Za dobu, kdy jsem pracoval s grafy, jsem si všiml některých vzorů. Pokud se snížil provoz, vzrostla rychlost vozidel. V jiném případě s přibývajícím počtem vozidel klesla průměrná rychlost, což si vysvětluji nějakou překážkou na dálnici (například havárií), nebo vznikem kolony (v návaznosti na havárii nebo přílišným počtem vozidel). Případu, kdy vozidel na dálnici ubylo, ale přesto došlo k prudkému zpomalení, by mohl odpovídat například obchvat nebo stavební práce na vozovce.

```
private void AddLayerAdvanced()
{
    ++counter;

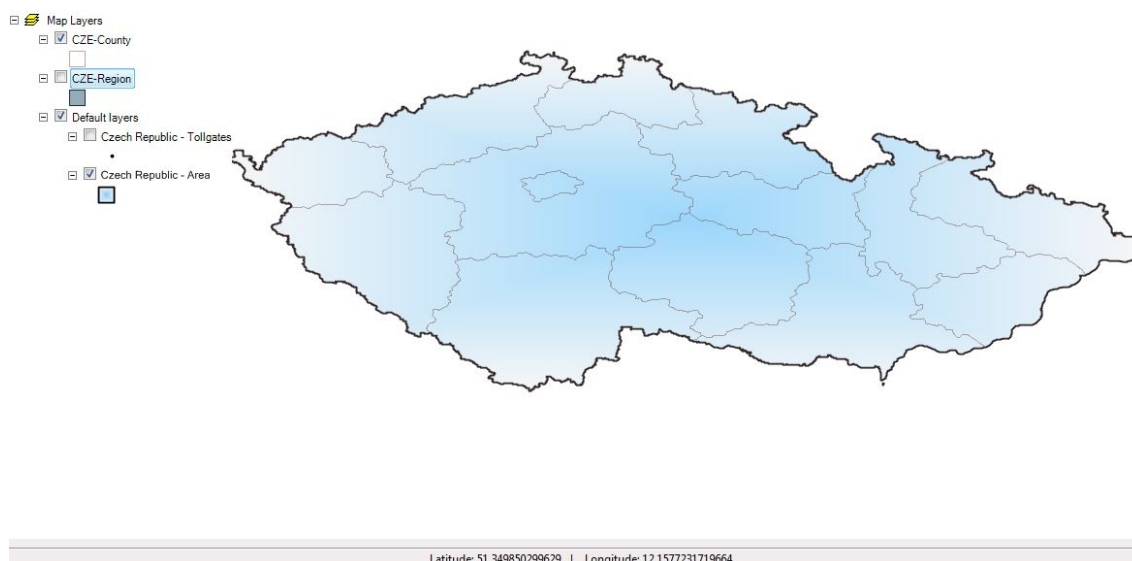
    IMapLayer myLayer = uxMap.AddLayer();
    CustomLayers.Add(myLayer);
    MapGroup customGroup = new MapGroup(uxMap, "Custom layers");

    try
    {
        foreach (var layer in CustomLayers)
        {
            customGroup.Layers.Add(layer);
            uxMap.Layers.Remove(layer);
        }
    }

    catch (Exception ex)
    {
        if (!(ex is NullReferenceException))
        {
            uxMap.Layers.RemoveAt(1);
        }
    }

    finally
    {
        if (counter > 1)
            uxMap.Layers.RemoveAt(1);
    }
}
```

Obrázek 41: přidání shapefilu do mapy jako vrstvu ve skupině Custom layers



Obrázek 42: Možné zobrazení České Republiky za pomoci přidanych vrstev

```
private void uxMap_MouseMove(object sender, MouseEventArgs e)
{
    DotSpatial.Topology.Coordinate coordinates =
        uxMap.PixelToProj(e.Location);

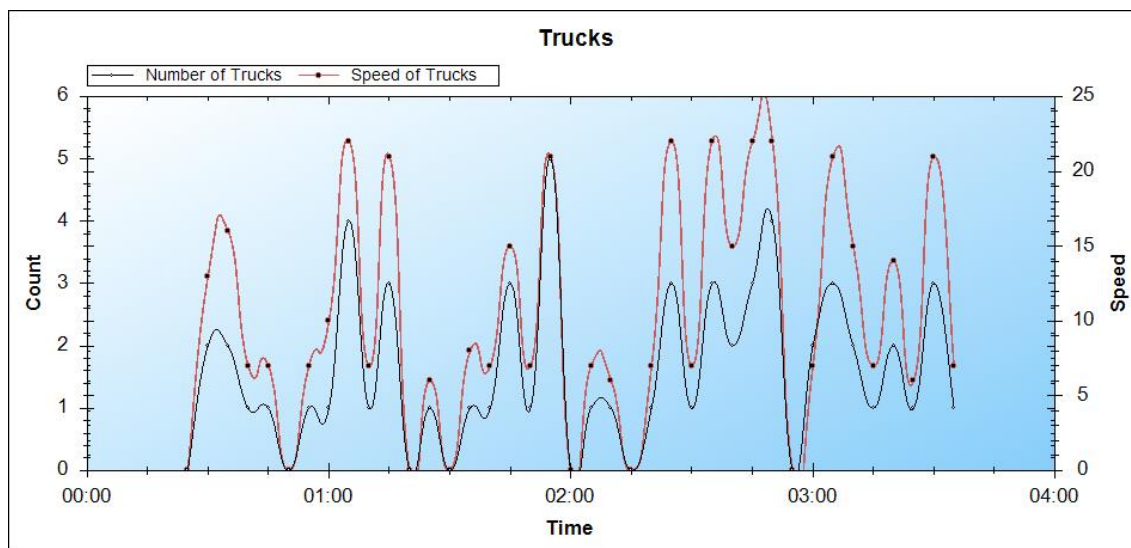
    mycord.mylat = coordinates.Y.ToString();
    mycord.mylong = coordinates.X.ToString();
}
```

Obrázek 43

```
try
{
    string longitude = mycord.myReplace(mycord.mylong);
    string latitude = mycord.myReplace(mycord.mylat);

    mylist.Add(mycord);
    uxComboBoxCoordinates.Items.Add(latitude + " , " + longitude);
}
```

Obrázek 44



Obrázek 45: závislost rychlosti nákladních vozů na jejich množství

9.5 Na co je třeba si dávat pozor

Narazil jsem na několik problémů, u kterých nebylo hned zprvu jasné, co je způsobilo.

Pokud se stane, že metoda `Graphics.DrawPath()` vyhodí výjimku `OutOfMemoryException`, příčina bude pravděpodobně souviset s nemožností určení směru u linií, které mají stejný počáteční a konečný bod.

Pro transformaci je možné použít rozhraní `GDI+`, což znamená, že souřadnice nemusíme pokaždé přepočítávat. Problém nastává v době, kdy je třeba použít datový typ `double`, jelikož třída pro transformaci požaduje datový typ `float`. Tato redukce snížila přesnost asi o 11 metrů.

10 Závěr

Závěr

Tato práce si kladla za cíl seznámit se s prostorovými daty a způsoby jejich vizualizace v prostoru za pomoci vhodných nástrojů a algoritmů pro vizualizaci dopravních analýz. Dále na základě naimplementované aplikace pro výkonnostní testy ověřit vhodnost těchto algoritmů či případně algoritmů vlastních a naimplementovat nástroj pro vizualizaci dopravních dat. V rámci zohlednění výsledků provedených výkonnostních testů bylo vybráno řešení, které vedlo k dokončení implementace funkční desktopové aplikace sloužící k vizualizaci dopravních dat s využitím prostorové databáze.

Přestože považuji svou práci za přínosnou, jsem si jistý, že na ni lze v budoucnu navázat a vylepšit ji například převedením do webové aplikace a doplněním o predikci vývoje dopravy na základě vzorů získaných z nasbíraných dat a grafů. Samotná aplikace již je předpřipravena k webové prezentaci, jelikož obsahuje i webové komponenty, které jsem nakonec nevyužil. Na užitečnosti by určitě přidala možnost sledovat záznamy pohybu vozidel na určitém segmentu v reálném čase nebo krátkém časovém úseku. To však, stejně jako rozšíření sledované oblasti za hranice České Republiky, závisí především na dodavatelích dat a technickém vybavení stroje, na kterém by data byla uložena a služba obstarávána.