Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra geomatiky

Porovnání vybraných databázových systémů s podporou prostorových dat

Bakalářská práce

Petr Trnka

Vedoucí práce:
Ing. František Kolovský

Plzeň, 2020

Prohlášení

Prohlašuji, že jsem bakalářskou práci na téma „Porovnání vybraných databázových systémů s podporou prostorových dat“ vypracoval samostatně. Veškeré zdroje, prameny a literaturu, z nichž jsem v práci čerpal, řádně cituji s uvedením odkazu na zdroj.

V Plzni ....

Petr Trnka

Poděkování

Na tomto místě bych rád poděkoval vedoucímu bakalářské práce Ing. Františku Kolovskému za věcné připomínky, cenné infromace a vstřícnost při konzultacích a vypracování bakalářské práce. Dále bych rád poděkoval Ing. Karlu Janečkovi Ph.D. za konzultaci k teoretickému základu práce. Mé poděkování patří i pracovníkům Centra informatizace a výpočetní techniky Západočeské univerzity v Plzni za pomoc při zřízení a správě serveru, na němž byla bakalářská práce zpracovávána. Na konec bych rád poděkoval svojí rodině za podporu při tvorbě této práce.

Abstrakt

Tato bakalářská práce se věnuje porovnání vybraných databázových systémů s podporou prostorových dat s důrazem na porovnání prostorových indexů. Porovnání bylo řešeno implementací sady testů, které reprezentují velmi často používané dotazy pro prostorová data. V práci je zhodnocení jednotlivých testů se statistickou analýzou výsledků. Na základě zjištěných údajů je možné pohlížet na MongoDB jako na dobrou alternativu k tradičním řešením.

Klíčová slova

systém řízení báze dat, prostorový index, PostgreSQL, MongoDB, CouchDB

Abstract

This bachelor thesis deals with the comparison of selected database systems with the support of spatial data with emphasis on the comparison of spatial indexes. The comparison was solved by implementing a set of tests that represent very frequently used queries for spatial data. The work include the evaluation of individual tests with statistical analysis of results. Based on the obtained data, MongoDB can be seen as a good alternative to traditional solutions.

Key words

Database Management System, spatial index, PostgreSQL, MongoDB, CouchDB

Obsah

Kapitola 1

Úvod

V dnešní době dochází k dynamickému rozvoji informačních technologií, se kterými souvisí i rozvoj v oblasti geografických informačních systémů. Z tohoto důvodu vznikla potřeba spravovat prostorová data v databázích a zároveň potřeba vyhodnocovat vhodnost jednotlivých systémů řízení báze dat pro práci s prostorovými daty.

V této práci byly porovnány vybrané systémy z hlediska výkonu na základě sady testů. Tyto testy dobře reprezentují výkon v oblasti prostorových dotazů využívající prostorové indexování.

Nejdůležitějším bodem práce bylo naimplementovat soubor testů pro každý systém s cílem porovnat výkon tradičního řešení v podobě objektově-relačního systému řízení báze dat se zástupci dokumentových systémů řízení báze dat. Mezi reprezentanty dokumentových systémů byly vybrány MongoDB a CouchDB, objektově-relační přístup reprezentuje tradiční řešení PostgreSQL s nadstavbou PostGIS. Důraz byl kladen zejména na potřebu provést porovnání mezi volně dostupnými systémy řízení báze dat. Během zpracování práce došlo k problémům s implementací testů pro CouchDB. Ukázalo se, že CouchDB není příliš intuitivní pro práci s prostorovými daty, protože implementace testů je obtížná.

Teoretická část práce pojednává o vybraných systémech řízení báze dat a zároveň popisuje vybrané indexační struktury, které jsou používány pro práci s prostorovými daty ve vybraných databázích. Praktická část práce popisuje tvorbu testů a výběr testovacích dat, která byla využita pro měření výkonu SŘBD.

Kapitola 2

Teoretické pozadí

Tato kapitola popisuje problematiku prostorového indexování a indexačních struktur, jež jsou použity pro prostorová data ve vybraných databázových systémech.

2.1 Prostorový index

Prostorový index je datová konstrukce umožňující efektivní přístup k jednomu či více záznamům v databázi na základě vyhledávacího klíče. Klíčem rozumíme podmnožinu atributů, na níž jsou funkčně závislé ostatní atributy relace. V případě prostorových dat chápeme jako klíč geometrii záznamu. Prostorový index umožňuje uživateli optimalizovat vyhledávání záznamů v databázi. Bez prostorového indexu by každé vyhledávání vyžadovalo sekvenční postupné prohledávání po záznamech, což by vedlo k delšímu zpracování dotazů. V následující kapitole jsou popsány prostorové indexy, které jsou implementovány ve vybraných systémech báze řízení dat.

2.2 R-strom

R-strom (R-tree) je prostorová datová struktura, kterou navrhl Antonin Guttman v roce 1984. R-strom představuje modifkaci B-stromu.

B-strom je prostorová datová struktura, v níž jsou uložená data již setříděna. B-strom je strom řádu M, kde M vyjadřuje maximální počet záznamů, které se vejdou do jednoho uzlu. Tento druh stromu obsahuje všechny listy na stejné úrovni (ve stejné hloubce stromu). Pro vnitřní uzly mimo kořene platí minimální naplněnost $\lceil \frac{M}{2} \rceil$ a maximální naplněnost M. B-strom je díky výše

zmíněným vlastnostem vyvážený a představuje ideální datovou strukturu pro situace, kdy jsou některá data uložena v jiném uložišti. Procházení dat uložených v této stuktuře pak probíhá v logaritmickém čase. [Bayer, 1972]

Záznamy v listových uzlech R-stromu obsahují ukazatele k datovým objektům, které reprezentují konkrétní prostorové objekty. R-strom používá pro ukládání objektů techniku minimálního ohraničujícího obdélníku (MOO) pro dimenzi $k = 2$, či techniku minimální ohraničující kostky (MOČ) pro dimenzi $k = 3$, které umožňují obalit objekt nebo skupinu objektů. Každý uzel R-stromu může obsahovat různý počet záznamů v závislosti na prostorovém umístění objektů. Je tedy potřeba definovat minimální a maximální naplnění uzlu z důvodu zachování vyváženosti stromu. R-strom je založen na heuristické optimalizaci s cílem minimalizovat velikost každého prostoru I, který se vyskytuje ve vnitřních uzlech. [Guttman, 1984]

Jak již bylo zmíněno, indexace objektů je založena na definování minimální ohraničující kostky (MOO/MOČ), která je souborem intervalů:

$$I = (I_0, I_1, ..., I_{k-1}) \tag{2.1}$$

kde k označuje dimenzi prostoru a I_i interval [a,b] popisující ohraničení objektu v dimenzi i. Pro použitá data v této práci je $i = 2$, protože využíváme 2D data a je tedy potřeba 4 parametrů.

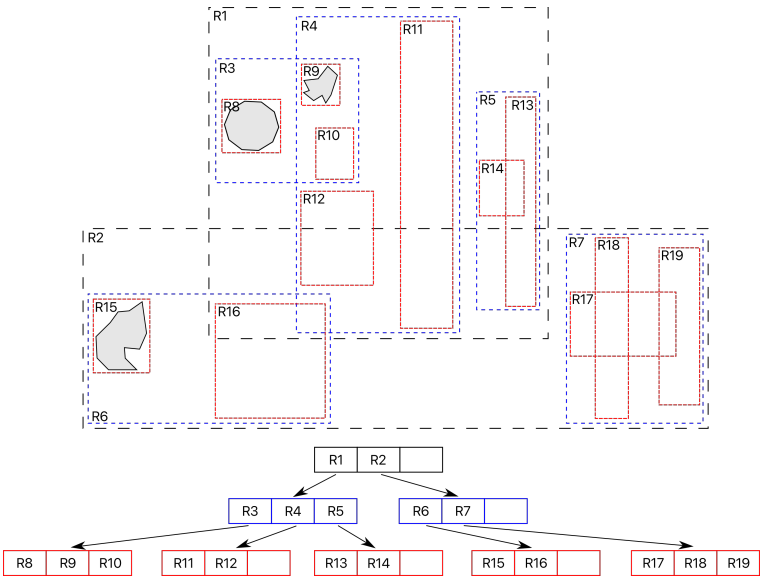
Struktura listu je dána pomocí dvojice: (I, Id), kde Id je identifikátor objektu.

Struktura vnitřního uzlu je dána pomocí dvojice: $(I, ukazatel)$, kde ukazatel ukazuje na podstrom R-stromu, pro nějž platí, že I pokrývá veškerá MOO, které v něm vyskytují.

Na rozdíl od B-stromu nemusí být dodrženo pravidlo o polovičním naplnění uzlů v nejhorším případě. Na obrázku 2.1 je vidět schéma R-stromu na jednoduchém příkladu.

Nechť M je maximální počet záznamů, které se vejdou do jednoho uzlu, a nechť $m \leq \frac{M}{2}$ je parametr určující minimální počet záznamů v uzlu R-stromu (R-strom je m-ární). R-strom pak splňuje tyto vlastnosti [Guttman, 1984]:

(1) Každý nelistový uzel má n bezprostředních následníků, $n \subseteq <m, M>$ (současně platí $m \leq \frac{M}{2}$).



Obrázek 2.1: R-strom

(2) Každý listový uzel obsahuje n indexových záznamů $n \subseteq <m, M>$.

(3) Kořen má nejméně dva bezprostřední následníky, není-li listem.

(4) Všechny cesty v R-stromech jsou stejně dlouhé.

(5) Všechny listy jsou na stejné úrovni.

Výška stromu obsahujícího N indexových záznamů je nanejvýš rovna: $h = \lceil log_m N - 1 \rceil$ a maximální počet uzlů při této výšce je roven [Guttman, 1984]:

$$\sum_{n=1}^{h} \frac{N}{m^k} = \frac{N}{m} + \frac{N}{m^2} + ... + 1.$$

R-strom je struktura dynamická, to znamená, že je založena na štěpení a slévání stránek. Vzhledem k tomu, že se mohou MOO v jednotlivých podstromech překrývat, je možná existence více než jedné možnosti, jak pokračovat v prohledávání z jednoho uzlu. Z tohoto důvodu je potřeba R-strom procházet vícekrát a prohledávání je složitější, proto je nutné provádět optimalizaci [Pokorný, 2000, Guttman, 1984, Zhang, 2017].

2.2.1 Algoritmus prohledávání R-stromu

Vstupem je kořen T a dotazovaný obdélník S. Výstupem algoritmu je pak množina nalezených záznamů.

(1) Jestliže T není list, pak pro každého potomka E rozhodneme, zda se jeho MOK protíná s S. Pro všechny protínající se potomky zavoláme tuto metodu na strom, jehož kořenem je právě tento potomek.

(2) Jestliže je T list, pak pro všechny potomky E určíme, zda se jejich MOK protíná s S. Pokud ano, pak přidáme tento záznam do množiny výsledků.

Mezi výhody R-stromu můžeme zařadit jeho širokou univerzálnost.

Mezi nevýhody R-stromu řadíme již uvedenou nejednoznačnost při prohledávání stromu. Provedení dotazu na umístění objektu může vést k prozkoumávání více cestami od kořene po úroveň listu, hledání tedy není určeno jedinou větví. Tato vlastnost je způsobena překrýváním minimálních ohraničujících kostek v jednotlivých podstromech, což může mít za následek zhoršení výkonu, zejména pokud je překrytí MOČ významné. Další nevýhoda souvisí s překrytím MOČ. Pokud dojde k vyšší míře překrytí u velkých MOČ, pak může dojít ke snížení výkonu při dotazování na rozsah z důvodu velkého množství prázdného prostoru. [Bayer, 1972]

2.2.2 GiST index

Zástupcem datové struktury typu R-strom je GiST (Generalizovaný prohledávací strom). GiST je rozšiřitelná datová struktura umožňující rozvíjet indexy nad různými druhy dat a zároveň podporuje jakékoli vyhledávání těchto dat. Tento balíček sjednocuje řadu populárních vyhledávacích stromů do jedné datové struktury (např.: R-strom, B^+-strom), což eliminuje potřebu vytváření více vyhledávácích stromů. Kromě sjednocení všech těchto struktur má GiST na rozdíl od zmíněných stromů jednu klíčovou funkci, a to rozšiřitelnost dat i dotazů [Hellerstein, 2004].

GiST index vždy obsahuje klíč a ukazatel na data v listech (hraniční uzly stromové struktury). Dále obsahuje tvrzení, které je u R-stromu ohraničující obdélník obsahující všechny body dostupné z vnitřního uzlu. Ukazatel představuje propojení na potomky ve vnitřních uzlech stromové struktury. Ukazatel a tvrzení označujeme jako záznam indexu, přičemž každý uzel může obsahovat více těchto záznamů indexu [Štěhule, 2008].

GiST index je definován následujícími operacemi:

1. **Operace nad klíči** - tyto metody jsou specifické pro danou třídu objektů a určují konfiguraci GiST indexu (Key Methods):
   - Konzistence (Consistent) - Jestliže je zaručeno v záznamu indexu, že tvrzení vyhovuje dotazu s danou hodnotou, pak vrací logickou hodnotu false.
   - Sjednocení (Union) - Pro zadanou množinu záznmů indexu vrátí tvrzení platné pro všechny záznamy v množině.
   - Komprese (Compress) - Nastavuje vhodný formát pro fyzické uložení, u prostorových dat určuje hraniční trojúhelník.
   - Dekomprese (Decompress) - Načítá formát pro fyzické uložení, opak metody Komprese.
   - Cena (Penalty) - Metoda vrací hodnotu ve významu ceny na vložení nové položky do konkrétní části stromu.
   - Výběr dělení (PickSplit) - Metoda, která určuje které položky záznamu zůstanou na původní stránce v přídadě, že je nutné rozdělit stránku indexu.
   - Identita (Same) - Metoda vrací logickou hodnotu true v případě, že jsou porovnávané položky indentické.
2. **Operace nad stromem** - obecné operace, které volají operace nad klíči (Tree methods):
   - Vyhledávání (Search) - operace, která volá metodu „Konzistence“,
   - Vložení (Insert) - operace volá metodu „Cena“ a „Výběr“,
   - Mazání (Delete) - operace volá metodu uvKonzistence [Štěhule, 2008].

Výhodou GiST indexů je možnost vytvoření doménově specifických indexů vázaných na vlastní typy vývojářům znalým doménové oblasti bez toho, aby se nutně staly databázovými specialisty [Štěhule, 2008].

2.3 Diskrétní globální sítě

Diskrétní globální síť je prostorová datová struktura, která se skládá ze sady buněk. Tyto buňky zaplňují celý zemský povrch, nebo mohou zaplnit pouze oblast s geometrií prvku. Časté využití diskrétních globálních sítí

je s buněčnými oblastmi, které mají nepravidelný tvar a velikost. V klasickém případě jsou buňky čtvercového nebo obdélníkového tvaru, nejčastěji používanými pravidelnými diskrétními globálními sítěmi jsou ty, které jsou založeny na geografickém souřadnicovém systému (zeměpisná šířka - délka) [Sahr et al., 2003].

Výhodou je univerzálnost použití této struktury jak pro vektorová data, kde soubor bodů diskrétní globální sítě nahrazuje tradiční dvojice souřadnic, tak i pro rastrová data, kde každá oblast tvoří pixel [Sahr et al., 2003].

2.4 B$^+$-strom

B$^+$-strom je prostorová datová struktura, která představuje modifikaci B-stromu, který navrhl Rudolf Bayer spolu s Edwardem M. McCreightem.

B$^+$-strom sdílí stejné vlastnosti jako B-strom. B-strom je strom řádu M, který obsahuje všechny listy na stejné úrovni (ve stejné hloubce stromu). Pro vnitřní uzly mimo kořene platí minimální naplněnost $\lceil \frac{M}{2} \rceil$ a maximální naplněnost M. B-strom je díky výše zmíněným vlastnostem vyvážený a představuje ideální datovou strukturu pro situace, kdy jsou některá data uložena v jiném uložišti [Bayer, 1972]. Procházení dat uložených v této stuktuře probíhá stejně jako v případě B-stromu v logaritmickém čase. Na rozdíl od B-stromu jsou všechna data uložena pouze na listech. Kořen má nejvýše M potomků, minimální počet potomků není stanoven [Bayer, 1972].

B$^+$-strom obsahuje v listové struktuře kromě vlastních klíčů a identifikátorů objektu i ukazatel na následující list. Tento jeden ukazatel nijak dramaticky nezvyšuje paměťovou náročnost v rámci listu, ale výrazně zvšuje výkon. Propojení listů je obvykle zleva doprava, takto propojený seznam listů nazýváme sada sekvencí. Toto řešení je vhodné např.: pro sekvenční prohledávání [Comer, 1979].

Výhodou B$^+$-stromu je jeho schopnost rychlého vkládání, vyhledávání a mazání dat díky ukazatelům na následující sourozence. Naopak mezi nevýhody tohoto stromu můžeme řadit mírně vyšší paměťové nároky, které jsou ale převýšeny schopností rychlého řešení dotazu [Comer, 1979].

2.4.1 2dsphere index

2dsphere index kombinuje techniky diskrétních globálních sítí a B^+-stromu. Nejprve dochází k rozdělení povrchu Země na buňky v různých úrovních rozlišení. Poté se aplikuje B^+-strom pro indexování geografických prvků aproximovaných jako jedna nebo více buněk. To znamená, že index 2dsphere je omezen na přijímání prostorových dat geodetického souřadného systému a na výpočet geometrií na zemském povrchu. Jeho výhodou oproti 2d indexu je jeho použití i pro liniové a polygonové prvky [Xiang et al., 2016]

Při vytvoření indexu dochází nejprve k výběru buněk, které plně pokrývají geometrii daného prvku. Velikost buňky diskrétní sítě je dynamická v rozmezí 500 m až 100 km v závislosti na velikosti zakryté plochy. Každá buňka obsahuje index B^+-stromu společně s geometrií objektu, který lze snadno spočítat dle umístění na povrchu koule [Mongo, 2013].

2.4.2 2d index

2d index stejně jako 2dsphere index kombinuje techniky diskrétních globálních sítí a B^+-stromu. Jeho odlišností je ovšem indexování v rovině namísto koule v případě 2dsphere indexu. Dalším rozdílem je možnost indexování pouze bodových prvků, protože tento index se vytvoří pouze nad koordinovanými páry, neumožňuje tedy ukládat geometrii objektu ve formě obrovského pole [Mongo, 2020].

Kapitola 3

Databázové systémy

3.1 PostgreSQL

PostgreSQL je open-source objektově relační systém řízení báze dat, který používá a rozšiřuje jazyk SQL (Structure Query Language). Vývoj tohoto SŘBD sahá do roku 1986, kde byl vyvíjen na Kalifornské univerzitě v Berkeley v rámci projektu POSTGRES. Tento projekt navazoval na předchozí projekt Ingres, oproti tomuto projektu dochází k popisu vztahů a k definici datových typů. Dne 30. června 1994 došlo k vydání POSTGRES pod licencí typu MIT [Štěhule, 2012].
PostgreSQL kromě výše zmíněného GiST indexu obsahuje vestavěnou podporu pro velké spektrum indexačních struktur:

- běžné indexy B-stromů
- hashových tabulek,
- generalizované vyhledávací stromy (GiST),
- generalizované převrácené indexy (GIN),
- indexační struktury definované uživatelem.

3.1.1 PostGIS

V této práci používáme v rámci PostgreSQL nadstavbu PostGIS. V rámci této nadstavby je přidána podpora pro geoprvky. První verze byla vydána v roce 2001 pod licencí General Public Licenece. Od roku 2006 má implementovanou specifikaci Simple Features for SQL konzorcia Open Geospatial Consortium [Ramsey, 2008]. PostGIS zahrnuje tato rozšíření:

- geometrické typy, které jsou popsány v ISO 19125,
- prostorové predikáty pro určení interakcí geometrií,
- prostorové operátory pro operace geoprostorových sad
- prostorový index GiST.

PostgreSQL s rozšířením PostGIS je jediným zástupcem mezi vybranými systémy řízení báze dat, který má implementován specifikaci Simple Feature for SQL.

3.2 MongoDB

MongoDB je zástupcem open source dokumentového systému řízení báze dat. Řadíme ho mezi NoSQL databáze a využívá pro uložení objektů dokumentu formát JSON nebo GeoJSON v případě prostorových dat. MongoDB bylo původně vyvinuto společností 10gen. V roce 2009 se již MongoDB vyvíjeno jako open source.
Součástí MongoDB je i podpora prostorových dat, která je zabezpečena dvojicí indexů:

- 2dsphere index,
- 2d index [Mongo, 2020].

3.3 CouchDB

Apache CouchDB je dalším zástupcem dokumentového systému řízení báze dat. Řadíme ho mezi NoSQL databáze, pro uložení objektů využívá dokumentu ve formátu JSON a GeoJSON v případě prostorových dat. Výhodou CouchDB je jeho schopnost řešit problémy se ukladáním objemných dat pro složitější systémy [Pethuru, 2008].

Součástí CouchDB je i podpora prostorových dat, prostorový index je postaven na R-stromu.

3.4 Srovnání popularity databázových systémů

Naše vybrané databázové systémy jsme se pokusili mezi sebou porovnat, co se týče jejich použití či jejich popularity. V níže uvedené tabulce uvádíme po-

pularitu jednotlivých námi vybraných databázových systémů společně s celosvětově nejpopulárnějšími, Pro porovnání jsme vybrali žebříček od iniciativy DB-Engines. Iniciativa DB-Engines vydává každý měsíc žebříček systémů pro

Pořadí	**Systém řízení báze dat**	**Skóre**
1.	Oracle	1346,39
2.	MySQL	1275,67
3.	Microsoft SQL Server	1096,29
4.	**PostgreSQL**	**503,37**
5.	**MongoDB**	**421,12**
...	...	...
32.	**CouchDB**	**18,12**

Tabulka 3.1: Popularita systémů (prosinec 2019) [DB-Engines, 2019]

správu databází seřazených sestupně podle popularity. Popularita systémů se zjišťuje na základě vyvážené metodiky.

Kapitola 4

Zpracování

Tato kapitola popisuje postup zpracování dat, která byla použita pro testování ve vybraných systémech řízení báze dat.

4.1 Zdrojová data

Všechna zdrojová data pochází z projektu OpenStreetMap. Výhodami OpenStreetMap jsou otevřenost dat, kompatibilita s licencí Open Database License (ODbL) a jejich relativně velká hustota a podrobnost. Použitá data jsou vytvářena komunitou příspěvatelů OpenStreetMap.

Pro tuto práci byla použita podmnožina těchto dat, konkrétně byla vybrána oblast označená jako North America, která obsahuje území USA, Kanady, Mexika, Grónska a části Ruska (Čukotka). Obrázek 4.1 znázorňuje vymezení oblasti pomocí polygonu.

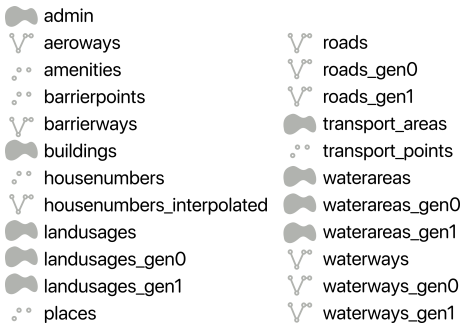
Obrázek 4.1: Oblast vybraných dat

Data byla získána pomocí stahovací služby Geofabrik, která umožňuje stahovat data ve formátu shapefile nebo osm.pbf. Extrakce dat byla provedena kde dni 23. října 2018, objem dat odpovídá ≈ 8,13 GB.

4.2 Imposm

V dalším kroku byla zdrojová data importována do databáze PostgreSQL. K tomu byl použit importovací nástroj Imposm. Tento nástroj umožňuje přečíst vybraný soubor osm.pbf a importovat obsažená data do vytvořené databáze. Kromě této funkce nabízí tento nástroj i aktualizaci vybraných dat.

Nástroj při importu v základní konfiguraci rozdělí data do celkem 23 tematických vrstev, přičemž každá z vrstev je uložena do jedné tabulky v databázi. Přehled vrstev včetně geometrického typu je uveden na Obrázku 4.2.



Obrázek 4.2: Tematické vrstvy z dat OpenStreetMap

V počtu 23 tematických vrstev je zahrnuto 15 vrstev negeneralizovaných a 8 vrstev generalizovaných. Mezi generalizovanými vrstvami se vyskytují 2 typy, které se liší volbou limitní vzdálenosti bodů:

- pro generalizaci gen0 je tolerance 200 m,
- pro generalizaci gen1 je tolerance 50 m.

4.3 Spojení vrstev

Pro potřeby testování proběhlo spojení již importovaných vrstev do 3 vrstev podle geometrických typů. V případě polygonových vrstev došlo k rozdělení

objektů typu MultiPolygony na jejich elementy, a to za účelem budoucího spojení polygonových vrstev. Z tohoto důvodu došlo k nárůstu počtu dat u polygonových vrstev, tuto změnu shrnuje Tabulka 4.1. Naopak u bodových a liniových vrstev nedošlo k žádnému rozdělení, neboť zdrojová data již obsahují bodové, případně liniové prvky.

Vrstva	**Počet prvků** Polygony a MultiPolygony	**Počet nových prvků** Polygony
admin	42 209	57 334
buildings	28 150 049	28 157 490
landusages	4 868 053	4 937 686
landusages_gen0	359 085	411 482
landusages_gen1	1 498 709	1 560 357
transport_areas	25 292	25 337
waterareas	4 713 625	4 718 511
waterareas_gen0	193 558	196 718
waterareas_gen1	876 454	880 756
Σ	**40 727 034**	**40 945 671**

Tabulka 4.1: Počet prvků v jednotlivých vrstvách

Celkový počet záznamů v tabulce bodů je 8 508 902, linií je 58 208 735 a polygonů 40 945 671.

4.4 Distribuce dat mezi vybranými SŘBD

V dalším kroku bylo potřeba již částečně upravená data distribuovat z databáze PostgreSQL do MongaDB a CouchDB. Obě zmíněné SŘBD jsou zástupci dokumentových databází, proto bylo potřeba vytvořit konverzi záznamů z relací. Jako nejvhodnější se ukázalo použití formátu GeoJSON, který je podporován v obou systémech. GeoJSON je otevřený standardní formát, který slouží k reprezentaci geografických prvků a jejich neprostorových atributů. Je založen na Java Object Notation (JSON).
Naplnění databází proběhlo pomocí konverzních programů pro každou z vrstev. Tyto programy spojily vybrané SŘBD s vytvořenou relací v PostgreSQL a zároveň zkonvertovaly geometrické objekty do struktury GeoJSON.

4.5 Validita dat

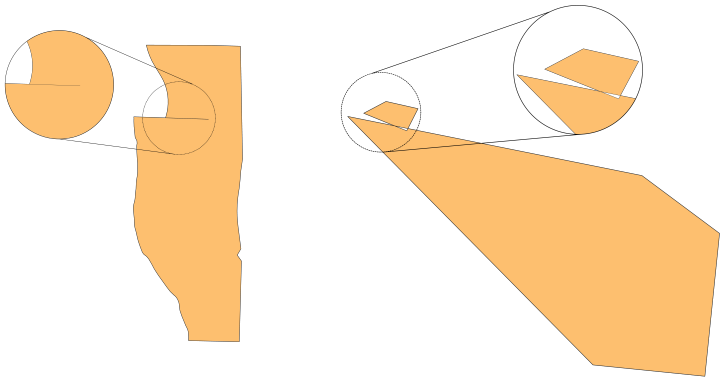
Jednou z nevýhod dat z OpenStreetMap je nevalidnost dat. Tato nevýhoda plyne již ze samotného způsobu tvorby dat. Tabulka 4.2 shrnuje výskyt chyb polygonových dat po jednotlivých tematických vrstvách.

Vrstva	**Počet záznamů**	**Počet chyb**	**Podíl validních záznamů** %
admin	57 334	2	≈ 100,00
buildings	28 157 490	0	100
landusages	4 937 686	36	≈ 100,00
landusages_gen0	411 482	2 834	≈ 99,31
landusages_gen1	1 560 357	190	≈ 99,98
transport_areas	25 337	0	100
waterareas	4 718 511	1	≈ 100,00
waterareas_gen0	196 718	6865	≈ 96,51
waterareas_gen1	880 756	807	≈ 99,91
souhrn	**40 945 671**	**10 735**	**≈ 99,97**

Tabulka 4.2: Validita polygonů v rámci jednotlivých vrstev

Při pohledu na rozdělení chyb v rámci vrstev můžeme vidět trend, kdy u generalizovaných vrstev dochází k několikanásobnému nárůstu počtu chyb oproti výchozí vrstvě, která byla na počátku generalizace. Tento nárůst je pravděpodobně způsoben nesprávnou generalizací nástrojem Imposm, při níž dochází k porušení validnosti objektů z daných vrstev. Na Obrázku 4.3 je vidět ukázka 2 zástupců nevalidních polygonů.

Validita dat byla řešena při zpracování až zpětným ověřením. Po vytvoření indexu v PostgreSQL nebyly odhaleny žádné chyby ve validitě dat, které by znemožňovaly vytvoření indexu. Po provedení postupu z kapitoly 4.4 a tvorbě 2dsphere indexu v databázi MongoDB ale došlo k chybovým hlášením, která byla způsobena nevalidností dat. Tyto chyby se vyskytují pouze ve vrstvě polygonů a jsou způsobeny špatnou generalizací nástrojem Imposm. Zároveň se tato chyba projevuje kvůli indexu 2dsphere, který indexuje objekty na sféře. Pro řešení tohoto problému bylo potřeba vytvořit další nástroj, který by odhalil chyby při samotném vytvoření indexu. Nejprve bylo potřeba vytvořit novou tabulku a nad ní vytvořit prostorový index 2dsphere. Poté bylo možné



(a) (b)

Obrázek 4.3: Ukázka nevalidních polygonů

opakovat totožný postup naplnění, v tomto případě rozšířený o nástroj na odhalení nevalidních polygonů. Po úspěšné distribuci bylo potřeba zjištěné nevalidní polygony odstranit i z tabulky v PostgreSQL, odkud jsou všechna data distribuována do dalších systémů.

Mimo tento postup selekce dat neproběhlo žádné další testování validity dat.

4.6 Topologie dat

Při zpracování dat nedošlo k nastavení žádných topologických pravidel, a to z důvodu příliš velkého objemu dat. Proběhl pokus o vytvoření topologických pravidel v softwaru ArcCatalog od firmy Esri, ale kvůli velkému objemu dat se nezdařil. Při řešení tohoto problému nepomohlo ani rozdělení vrstev zpět na dílčí vrstvy, které byly k dispozici již ze softwaru Imposm, a to kvůli ruční editaci chyb v topologickém pravidle.

Kapitola 5

Implementace

Tato kapitola popisuje postup implementace sady testů pro testování vybraných systému řízení báze dat z pohledu výkonu prostorových dotazů. Všechny testy byly implementovány v programovacím jazyce Java. Při implementaci bylo potřeba konstruovat dotazy tak, aby všechny byly závislé na prostorovém indexu.

Před samotnou implementací bylo potřeba vytvořit sadu testů, které by nejlépe porovnávaly vybrané systémy řízení báze dat. Pro tento účel byla vytvořena sada 5 testů obsahující dotazy, které jsou velmi časté při práci s prostorovými daty. Zároveň tyto dotazy mají za cíl ukázat různé úlohy, které testují vybrané systémy řízení báze dat, respektive testují vybrané indexační struktury z hlediska jejich vhodnosti pro daný případ použití.

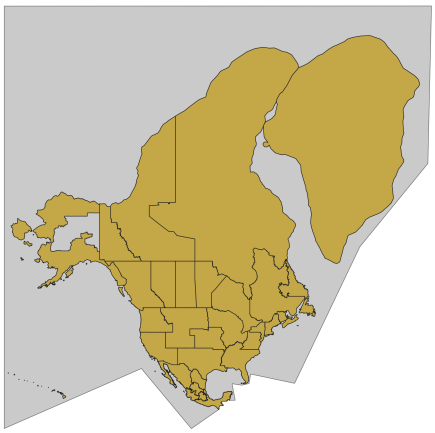
Vytvořená sada testů obsahují tyto úlohy:

- vložení záznamu,
- hledání nejbližšího souseda,
- hledání k-nejbližších sousedů,
- protínání prvků (Intersect),
- prostorové spojení (Spatial join).

5.1 Sada náhodných dat pro testování

Pro potřeby testování byla vytvořena sada náhodně vygenerovaných dat. Nejprve došlo k rozdělení zájmové oblasti do 33 bloků podle hranic regionů

vymezených jednotlivými státy v zájmové oblasti, což je znázorněno na Obrázku 4.1. Zároveň byly vynechány oblasti moří, kde není žádné nebo velmi řídké zastoupení prvků ve zdrojových datech.



Obrázek 5.1: Rozdělení zájmové oblasti na bloky

V blocích pak bylo zjištěno zastoupení jednotlivých prvků ve zdrojových datech. Poté byly na základě zjištěných zastoupení náhodně generována data tak, aby celkový počet prvků ve všech blocích byl roven 1000 a odpovídal hustotě prvků v oblasti. V tabulce 5.1 je uvedeno rozdělení prvků do bloků.

Dalších 10 prvků bylo generováno náhodně v zájmové oblasti, tyto prvky byly použity na začátku testu, tak aby se eliminovaly časové prodlevy při spuštění testu, které by mohly ovlivnit výsledný čas dotazu.

5.2 Vložení prvku

První z testů je vložení záznamu. Vložení probíhá vždy do jedné z vrstev podle typu prvku, který je vkládán. Tento test byl proveden pro všechny tři typy prvků, tedy pro body, linie a polygony. Pro každý prvek byl zjišťován čas, za který dojde k jeho vložení a zároveň zjišťujeme celkový čas testu. Na konci testu jsou tato vložená data smazána, aby nedocházelo ke zkreslení při použití stejných dat pro další testy se stejnou vrstvou.

Výsledkem testu je:

- čas zpracování dotazu pro jednotlivé vkládané záznamy,

Oblast	**Počet bodů**	**Počet linií**	**Počet polygonů**
Aljaška	1	3	7
Alberta	25	15	17
Britská Kolumbie	36	130	33
Grónsko	0	0	2
Havaj	1	1	1
Manitoba	2	12	41
MEX1	4	3	5
MEX2	1	1	2
MEX3	1	1	1
MEX4	2	2	2
MEX5	3	3	3
MEX6	3	4	4
Newfoundland	1	53	91
Nové Skotsko	31	13	7
Nový Brušvik	2	7	8
Nunavut	0	23	28
Ontario	178	110	86
Ostrov prince Eduarda	7	2	1
Québec	103	74	64
Rusko	0	0	0
Saskatchewan	2	8	16
Severozápadní teritorium	0	7	17
USA1	11	10	56
USA2	35	13	20
USA3	54	37	40
USA4	130	106	92
USA5	65	67	133
USA6	77	52	37
USA7	28	21	26
USA8	20	61	39
USA9	150	124	90
USA10	27	28	17
Yukon	0	9	14
Σ	**1000**	**1000**	**1000**

Tabulka 5.1: Rozdělení vygenerovaných dat do bloků

- celkový čas testu.

Test má za cíl otestovat rychlost vložení prvku do uzlu ve stromové indexační struktuře.

Na obrázcích C.1 a C.2 jsou vidět kódy dotazů pro vložení záznamu. Dotaz pro MongoDB a CouchDB je totožný, proto je uveden jen jednou.

5.3 Hledání nejbližšího souseda

Druhým testem v pořadí je hledání nejbližšího souseda. V tomto testu byly využity vygenerované body, k nimž byl hledán nejbližší soused z bodové, liniové a polygonové vrstvy. Test probíhá iterační metodou, na počátku testu je nastavena počáteční vzdálenost 10 metrů. Pro tuto vzdálenost zjišťujeme, zda se nachází nějaký soused ve vzdálenosti bližší než je nastavená hodnota. Pokud takový soused existuje, pak zjišťujeme, zda je jediný v této vzdálenosti. V případě, že by prvků do zvolené vzdálenosti bylo více, pak vybereme nejbližšího z nich. Pokud se v dopustné vzdálenosti nenachází žádný prvek, pak zvyšujeme počáteční vzdálenost desetinásobně a opakujeme postup.

Výsledkem testu jsou:

- dvojice: bod i a nejbližší prvek,
- čas zpracování dotazu pro jednotlivé body,
- celkový čas testu.

Na obrázcích C.3 a C.4 jsou vidět kódy dotazů pro hledání nejbližšího souseda.

5.4 Hledání k-nejbližších sousedů

Následující test je velmi podobný předchozímu testu hledání nejbližšího souseda. V tomto testu byly opět využity vygenerované body, k nimž bylo hledáno k-nejbližších sousedů z bodové, liniové a polygonové vrstvy. Empiricky byla zvolena hodnota k=10. Test probíhá iterační metodou, na počátku testu je nastavena počáteční vzdálenost 10 metrů. Pro tuto vzdálenost zjišťujeme, zda se nachází alespoň k sousedů ve vzdálenosti bližší než je nastavená hodnota. Pokud výsledkem dotazu je více než k prvků, pak vybereme k nejbližších.

V případě, že ve zvolené vzdálenosti se nachází prvky, ale jejich počet je nižší než k, pak zvyšujeme vzdálenost desetinásobně a opakujeme postup. Stejným postup použijeme pro případ, že by v zadané vzdálenosti nebyl žádný prvek.

Výsledkem testu jsou:

- dvojice: bod i a k-nejbližších prvků,
- čas zpracování dotazu pro jednotlivé body,
- celkový čas testu.

Na obrázcích C.5 a C.6 jsou vidět kódy dotazů pro hledání k-nejbližšího souseda.

5.5 Intersect

Cílem tohoto testu je otestovat výkon vybraného systému řízení báze dat z hlediska prostorového vztahu, zde konkrétně protínání. K jednomu zadanému prvku jsou vždy hledány všechny prvky z cílové vrstvy, které se se zadaným prvkem protínají. Pro tento test byly vybrány 4 kombinace:

- náhodně vygenerované body a polygony v databázi,
- náhodně vygenerované linie a linie v databázi,
- náhodně vygenerované linie a polygony v databázi,
- náhodně vygenerované polygony a polygony v databázi.

Předmětem tohoto testu je pro vygenerované prvky hledat prvky v databázích, s nimiž mají průnik.

Výsledkem testu jsou:

- dvojice: vygenerovaný prvek i a prvek z databáze, s nimž se protíná,
- čas zpracování dotazu pro jednotlivý prvek,
- celkový čas testu.

Na obrázcích C.7 a C.8 jsou vidět kódy dotazů hledání průniků.

5.6 Spatial join

Poslední test se částečně podobá předchozímu testu. V tomto testu zjišťujeme křížení/průnik mezi prvky, ale v tomto případě je hledáme mezi prvky vrstev v databázi. Při implementaci tohoto testu došlo k různé implementaci napříč vybranými systémy z důvodu obtížné implementace pro MongoDB, které není v případě tohoto testu přívětivé. Zároveň bylo pro potřeby testování nutné vybrat pouze podmnožinu dat z důvodu časové náročnosti testu, která převyšovala limit VPN připojení, který je na ZČU nastaven na 8 hodin. Test probíhal pouze pro vrstvu linií a polygonů, zjišťovaly se pouze vzájemné průniky mezi liniemi a mezi polygony. Pro oba testy byla vybrána podmnožina dat čítající 10 000 prvků.

Výsledkem testu jsou:

- dvojice: prvek i z databáze a prvek j z databáze, s nimž se protíná,
- celkový čas testu.

Na obrázku C.9 je vidět kód dotaz pro prostorové spojení. Pro MongoDB byla použita analogie kódu v obrázku C.8.

5.7 Hardwarové vybavení

Veškeré testy byly prováděny na notebooku Macbook Air s dvoujádrovým procesorem Intel(R) Core(TM) i5-5250U CPU @ 1.60 GHz s 8 GB RAM. Následně byly tyto testy provedeny na virtuálním stroji v cloudu CIV ZČU se 4 CPU a 16 GB RAM. Všechny výsledky, uveřejněné v kapitole 6, pochází z testování ve virtuálním stroji.

Kapitola 6

Výsledky

Tato kapitola shrnuje výsledky sady testů. Všechny výsledky testů jsou k dispozici na přiloženém nosiči nebo ve webové službě GitLab. Pro všechny dílčí časové výsledky byla vytvořena základní chrakteristiku statistického souboru, která zahrnuje výběrové charakteristiky polohy a variability včetně obecných a centrálních momentů.

6.1 Vložení záznamu

Na obrázcích B.1, B.2 a B.3 si můžeme prohlédnout výsledky testu vložení záznamu. Tento test je jediný, u nějž máme výsledky pro všechny vybrané systémy řízení báze dat. Pro lepší přehlednost hodnot v grafu byla zvolena logaritmická časová osa.

Z obrázku B.1 je vidět, že vložení bodu do vrstvy je nejrychlejší v případě databáze MongoDB, zástupce dokumentových systémů. Zároveň je potřeba zmínit, že nejpomaleji se vložila bodová data do CouchDB, druhého ze zástupců dokumentových databází.

Na obrázcích B.2 pro vložení linií a B.3 pro polygonů si můžeme všimnout změn, kdy nejrychlejší bylo vložení do databáze PostGIS, MongoDB bylo v rámci vložení linií těsně nejpomalejší a CouchDB byl jednozančně nejpomalejší v testu vložení polygonů. Pomalejší čas při vložení linií a polygonů do databáze MongoDB je nejspíše způsoben vyšší pamětovou náročností, která u těchto 2 testů převýšila schopnost databáze MongoDB rychle vkládat prvky.

Tabulka 6.1 obsahuje základní statistiku pro výsledky testu vložení záznamu.

SŘBD	Vložení bodů			Vložení linií			Vložení polygonů		
	t [s]	μ [ms]	σ [ms]	t [s]	μ [ms]	σ [ms]	t [s]	μ [ms]	σ [ms]
CouchDB	38,3	38,4	23,9	35,9	35,9	17,8	49,3	49,3	51,5
MongoDB	5,6	5,6	13,1	37,9	37,9	44,7	28,9	28,9	26,4
PostGIS	12,8	12,8	18,0	17,2	17,2	25,6	22,3	22,3	18,8

Tabulka 6.1: Statistiky testu vložení záznamu

Pozn. Textové soubory s dílčími výsledky pro PostGIS a MongoDB obsahují časy uvedené v μs.

6.2 Hledání nejbližšího souseda

Druhým testem v pořadí je hledání nejbližšího souseda. Na obrázcích B.4, B.5 a B.6 jsou zaznamenány výsledky pro MongoDB a PostGIS.

Při pohledu na tyto obrázky je vidět, že ve všech testech je rychlejší MongoDB než PostGIS. Zároveň vyplývá, že v testu hledání nejbližšího souseda pro liniovou vrstvu (Obrázek B.5) měl z počátku testu navrch PostGIS, ale přibližně kolem bodu č. 300 došlo k výraznému nárůstu času. Tento jev je způsoben zvýšením hustoty prvků v rámci testované vrstvy. V tomto testu se ukázalo, že použití funkce ST_Distance je pro seřazení bodů méně efektivní na úkor vlastností B^+-strom.

Dále lze vysledovat závislost času dílčího dotazu na počtu prvků ve vrstvě. V případě liniové vrstvy došlo k nárůstu času zpracování dílčího dotazu skoro stonásobně oproti bodové vrstvě a téměř osminásobně oproti polygonové vrstvě.

Tabulka 6.2 obsahuje základní statistiku pro výsledky testu hledání nejbližšího souseda.

SŘBD	NN bod			NN linie			NN polygony		
	t [s]	μ [ms]	σ [ms]	t [s]	μ [s]	σ [s]	t [s]	μ [ms]	σ [ms]
MongoDB	7,4	7,4	40,0	1084,3	1,1	13,9	234,3	234,3	533,9
PostGIS	29,4	29,4	193,4	2243,0	2,2	24,9	340,3	340,3	307,8

Tabulka 6.2: Statistiky testu hledání nejbližšího souseda

6.3 Hledání k-nejbližších sousedů

Na obrázcích B.7, B.8 a B.9 jsou zaznamenány výsledky testu hledání k-nejbližších sousedů pro MongoDB a PostGIS.

Při pohledu na tyto obrázky platí stejné závěry jako v případě hledání nejbližšího souseda. Na obrázku B.8 je vidět, že v testu hledání k-nejbližšího souseda pro liniovou vrstvu měl z počátku testu navrch PostGIS (stejně jako v předchozím testu), ale přibližně kolem linie č. 300 došlo k výraznému nárůstu času. Přičina je stejná jako v případě testu hledání nejbližšího souseda, ato použití funkce ST_Distance pro seřazení prvků.

Na obrázku B.9 je vidět skokové zrychlení vyhodnocení dotazu pro PostGIS pro body č. 150 až č. 250. Tato anomálie je způsobena vysokou hustotou náhodně generovaných bodů v oblasti, kde je sice vysoká hustota polygonů, ale jejich vzdálenost od zadaného bodu je kratší. Z tohoto důvodu není potřeba nastavovat dlouhé vzdálenosti pro hledání a tím pádem dojde ke zkrácení času pro vyhodnocení dotazu.

Zároveň lze vysledovat závislost času dílčího dotazu na počtu prvků ve vrstvě. V případě liniové vrstvy došlo k nárůstu času zpracování dílčího dotazu více než desetinásobně oproti bodové vrstvě. V tomto testu se ale prokázala paměťová složitost polygonových prvků, kvůli nimž bylo hledání k-nejbližšího souseda nejdelší.

Tabulka 6.3 obsahuje základní statistiku pro výsledky testu hledání k-nejbližšího souseda.

SŘBD	kNN bod			kNN linie			kNN polygony		
	t [s]	μ [ms]	σ [ms]	t [s]	μ [s]	σ [s]	t [s]	μ [s]	σ [s]
MongoDB	127,2	127,2	825,3	957,2	1,0	8,7	2415,1	2,4	18,2
PostGIS	446,9	446,9	3119,1	5410,5	5,4	40,0	7404,0	7,4	59,9

Tabulka 6.3: Statistiky testu hledání k-nejbližšího souseda

6.4 Intersect

Čtvrtým testem v pořadí je Intersect. Na obrázcích B.10, B.11, B.12 a B.13 jsou zaznamenány výsledky pro MongoDB a PostGIS.

Při pohledu na tyto obrázky je vidět velmi nejednoznačný výsledek v testu na průnik vygenerované bodové vrstvy a polygonové vrstvy. Z obrázku B.10 je patrné, že v 1. části testu měl navrch PostGIS, ale kolem bodu č. 300 došlo k nárůstu času vyhodnocení dílčího dotazu vlivem zvyšující se hustoty polygonů v dané oblasti. Naopak trend času vyhodnocení dílčího dotazu pro MongoDB byl opačný, kdy se v druhé části testu projevil vliv rychlého zpracování B^+-stromu.

V obrázku B.11, který ukazuje výsledky pro test na průnik vygenerované liniové vrstvy s liniovou vrstvou, je vidět kolem linie č. 400 skok v délce vyhodnocení dílčího dotazu. Přičinou tohoto skoku, který se projevil u obou systémů, je zvyšující se hustoty linií v dané oblasti.

Mimo testu na průnik vygenerované bodové vrstvy a polygonové vrstvy se u všech ostatních testů projevil vysoký výkon databáze MongoDB, kdy ve všech zbývajících testech došlo ke trojnásobnému snížení celkového času, než tomu bylo při zpracování v databázi PostGIS.

Celkové časy a základní statistiku pro testy Intersect jsou shrnuty v tabulkách 6.4 a 6.5.

SŘBD	Bod-Polygon			Linie-Linie		
	t [s]	μ [ms]	σ [ms]	t [s]	μ [s]	σ [s]
MongoDB	185,1	185,1	510,5	1264,1	1,3	2,1
PostGIS	170,0	170,0	210,3	3391,0	3,4	6,3

Tabulka 6.4: Statistiky testu hledání průniku

SŘBD	Linie-Polygon			Polygon-Polygon		
	t [s]	μ [s]	σ [s]	t [s]	μ [ms]	σ [ms]
MongoDB	1174,8	1,2	2,0	368,7	368,7	641,2
PostGIS	3073,4	3,1	6,0	978,8	978,8	1748,7

Tabulka 6.5: Statistiky testu hledání průniku

6.5 Spatial join

Závěrečným testem sady testů bylo prostorové spojení. Na rozdíl od předchozích testů není pro tento test výsledek pro každý dílčí dotaz, a to z důvodu stavby dotazu, kdy je možné změřit pouze celkový čas.

Z tabulky 6.6 je vidět, že obou testech byl rychlejší PostGIS. Delší doba zpracování dotazu v systému MongoDB byla způsobena větší paměťovou náročností. Zároveň mohl výsledek dotazu ovlivnit odlišný způsob implementace.

SŘBD	Linie-Linie	Polygon-Polygon
	t [s]	t [s]
MongoDB	284,5	230,7
PostGIS	225,0	207,2

Tabulka 6.6: Statistiky testu prostorové spojení

Pro větší přehlednost byly výsledky všech testů sloučeny do souhrné tabulky 6.7.

*Pozn. Symbol * označuje hodnoty, které nebyly ve zpracovaném testu určeny.*

SŘBD	CouchDB			MongoDB			PostGIS		
	t [s]	μ [ms]	σ [ms]	t [s]	μ [ms]	σ [ms]	t [s]	μ [ms]	σ [ms]
Vložení bodů	38,3	38,3	23,9	5,6	5,6	13,1	12,8	12,8	18,0
Vložení linií	35,9	35,9	17,8	37,9	37,9	44,7	17,2	17,2	25,6
Vložení polygonů	49,3	49,3	51,5	28,9	28,9	26,4	22,3	22,3	18,8
NN bod	-	-	-	7,4	7,4	40,0	29,4	29,4	193,4
NN linie	-	-	-	1084,3	1084,3	13894,3	2243,0	2243,0	24925,2
NN polygon	-	-	-	234,3	234,3	533,9	340,3	340,3	307,8
kNN bod	-	-	-	127,2	127,2	825,3	446,9	446,9	3119,1
kNN linie	-	-	-	957,2	957,2	8736,9	5410,5	5410,5	40000,0
kNN polygon	-	-	-	2415,1	2415,1	18177,0	7404,0	7404,0	59849,1
Průnik bod-polygon	-	-	-	185,1	185,1	510,5	170,0	170,0	210,3
Průnik linie-linie	-	-	-	1264,1	1264,1	2147,8	3391,0	3391,0	6318,5
Průnik linie-polygon	-	-	-	1174,8	1174,8	1964,3	3073,4	3073,4	6036,3
Průnik polygon-polygon	-	-	-	368,7	368,7	641,2	978,8	978,8	1748,7
Prostorové spojení linie-linie	-	-	-	284,5	*	*	225,0	*	*
Prostorové spojení polygon-polygon	-	-	-	230,7	*	*	207,2	*	*

Tabulka 6.7: Souhrná tabulka s výsledky testů

Kapitola 7

Závěr

V práci byla implementována sada testů, na jejichž výsledku bylo možné vyvodit doporučení pro použití vybraného systému řízení báze dat.
Testování prokázalo zajímavý potenciál MongaDB, které se ukázalo jako dobrá alternativa z oblasti open-source. Na základě výsledků můžeme doporučit MongoDB jako vhodný systém pro použití dotazu na hledání nejbližšího souseda, hledání průniků mezi prvky vrstev a prostorové spojení. Výsledky práce korespondují se závěry článku [Coskun et al., 2019], který potvrzuje teorii o vhodnosti využití systému MongoDB v testu hledání nejbližších sousedů. Zástupce objektově-relačního systému řízení báze dat PostGIS je na základě výsledků vhodný v případě častého vkládání linií a polygonů a také pro použití při dotazu na hledání průniku bodů s polygony.

Jedním ze závěru práce je i odhalení chyby importovacího nástroje Imposm při generalizaci polygonových vrstev při zpracovní zdrojových dat. Tato skutečnost bude prodiskutována s vývojáři nástroje s cílem eliminovat tuto chybu pro budoucí použití.

Během provádění testů se vyskytlo mnoho nemalých problémů s validitou testovacích dat a s objemem testovacích dat. Dalším problémem byla implementace testů pro CouchDB, vetšina pokusů dopadla z důvodu obtížnosti negativně.

Literatura

[Bayer, 1972] Bayer, R. McCreight, E. (1972). Organization and maintenance of large ordered indexes. *Acta informatica*, 1:173–189.

[Comer, 1979] Comer, D. (1979). Ubiquitous b-tree. *ACM Comput. Surv.*, 11(2):121–137.

[Coskun et al., 2019] Coskun, I., Sertok, S., and Anbaroglu, B. (2019). K-nearest neighbour query performance analyses on a large scale taxi dataset: Postgresql vs. mongodb. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4213:1531–1538.

[DB-Engines, 2019] DB-Engines (2019). Method of calculating the scores of the db-engines ranking. [Online; citováno 29-Duben-2019].

[Guttman, 1984] Guttman, A. (1984). R-trees: A dynamic index structure for spatial searching. *Proceedings of the 1984 ACM SIGMOD international conference on Management of data - SIGMOD '84.* [Online; citováno 2-Únor-2020].

[Hellerstein, 2004] Hellerstein, J. M. (2004). The gist indexing project. [Online; citováno 30-Duben-2019].

[Mongo, 2013] Mongo (2013). New geo features in mongodb 2.4. [Online; citováno 9-Květen-2020].

[Mongo, 2020] Mongo (2020). 2d indexes. [Online; citováno 9-Květen-2020].

[Pethuru, 2008] Pethuru, Raj Ganesh, C. D. (2008). A deep dive into nosql databases: The use cases and applications. *Advances in computers*, 109:400.

[Pokorný, 2000] Pokorný, J. (2000). Prostorové datové struktury a jejich použití k indexaci prostorových objektů. *GIS Ostrava*.

[Ramsey, 2008] Ramsey, P. (2008). Postgis history. [Online; citováno 9-Květen-2020].

[Sahr et al., 2003] Sahr, K., White, D., and Kimerling, A. J. (2003). Geodesic discrete global grid systems. *Cartography and Geographic Information Science*, 30(2):121–134.

[Xiang et al., 2016] Xiang, L., Huang, J., Shao, X., and Wang, D. (2016). A mongodb-based management of planar spatial data with a flattened r-tree. *International Journal of Geo-Information*, 5(7):119.

[Zhang, 2017] Zhang, X. Du, Z. (2017). Spatial indexing. *The Geographic Information Science & Technology Body of Knowledge (4th Quarter 2017 Edition)*. [Online; citováno 30-Duben-2019].

[Štěhule, 2008] Štěhule, P. (2008). Postgis pro vývojáře. [Online; citováno 9-Květen-2020].

[Štěhule, 2012] Štěhule, P. (2012). Historie projektu postgresql. [Online; citováno 9-Květen-2020].

Příloha A

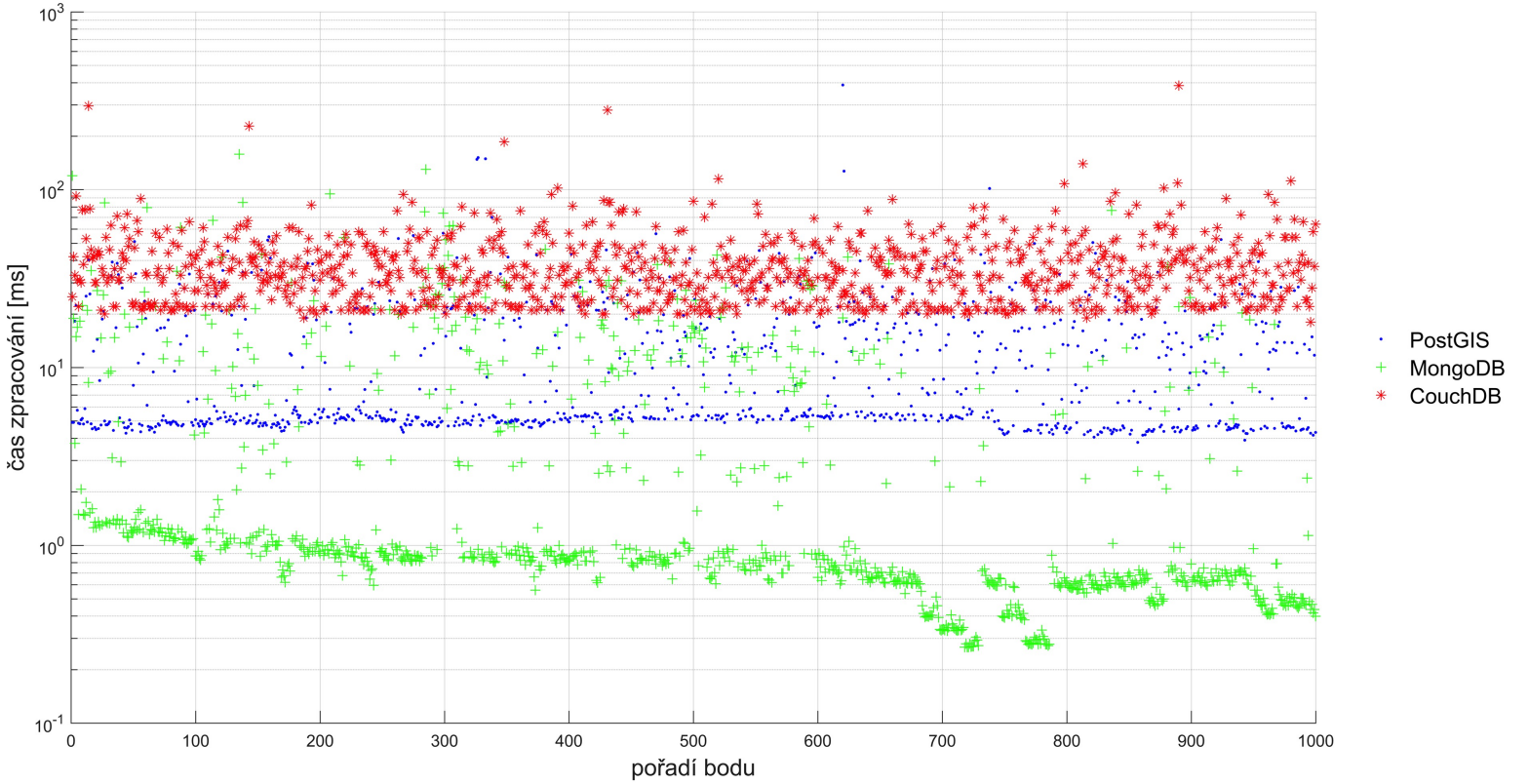
Obsah přiloženého nosiče

- `Trnka_BP.pdf` - bakalářská práce
- `grafy` - složka s grafy
- `ostatní` - složka s ostatními programy
- `readme.txt` - popis a návod k použití
- `statistiky` - složka se statistikami výsledků
- `testy` - složka s implementovanými testy
  - `Insert_Line_CouchDB.java`
  - `Insert_Line_Mongo.java`
  - `Insert_Line_Postgis.java`
  - `Insert_Point_CouchDB.java`
  - `Insert_Point_Mongo.java`
  - `Insert_Point_Postgis.java`
  - `Insert_Polygon_CouchDB.java`
  - `Insert_Polygon_Mongo.java`
  - `Insert_Polygon_Postgis.java`
  - `Intersect_Line_Line_Mongo.java`
  - `Intersect_Line_Line_Postgis.java`
  - `Intersect_Line_Polygon_Mongo.java`
  - `Intersect_Line_Polygon_Postgis.java`
  - `Intersect_Point_Polygon_Mongo.java`
  - `Intersect_Point_Polygon_Postgis.java`
  - `Intersect_Polygon_Polygon_Postgis.java`

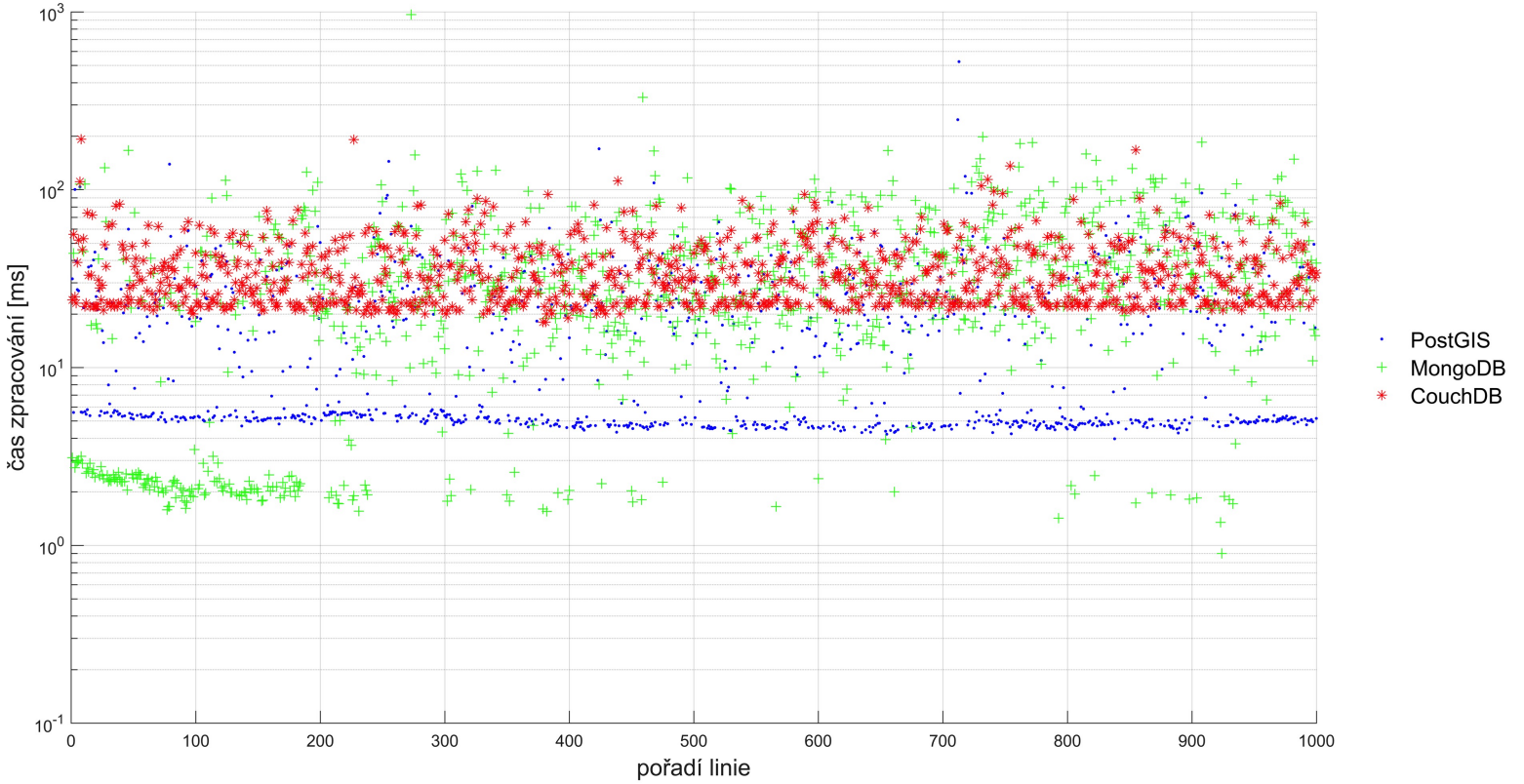
  - Intersect_Polygon_Polygon_Mongo.java
  - KNN_Line_Mongo.java
  - KNN_Line_Postgis.java
  - KNN_Point_Mongo.java
  - KNN_Point_Postgis.java
  - KNN_Polygon_Mongo.java
  - KNN_Polygon_Postgis.java
  - NN_Line_Mongo.java
  - NN_Line_Postgis.java
  - NN_Point_Mongo.java
  - NN_Point_Postgis.java
  - NN_Polygon_Mongo.java
  - NN_Polygon_Postgis.java
  - Spatial_join_Line_Line_Mongo.java
  - Spatial_join_Line_Line_Postgis.java
  - Spatial_join_Polygon_Polygon_Mongo.java
  - Spatial_join_Polygon_Polygon_Postgis.java
- `výsledky` - složka s výsledky testů

Příloha B

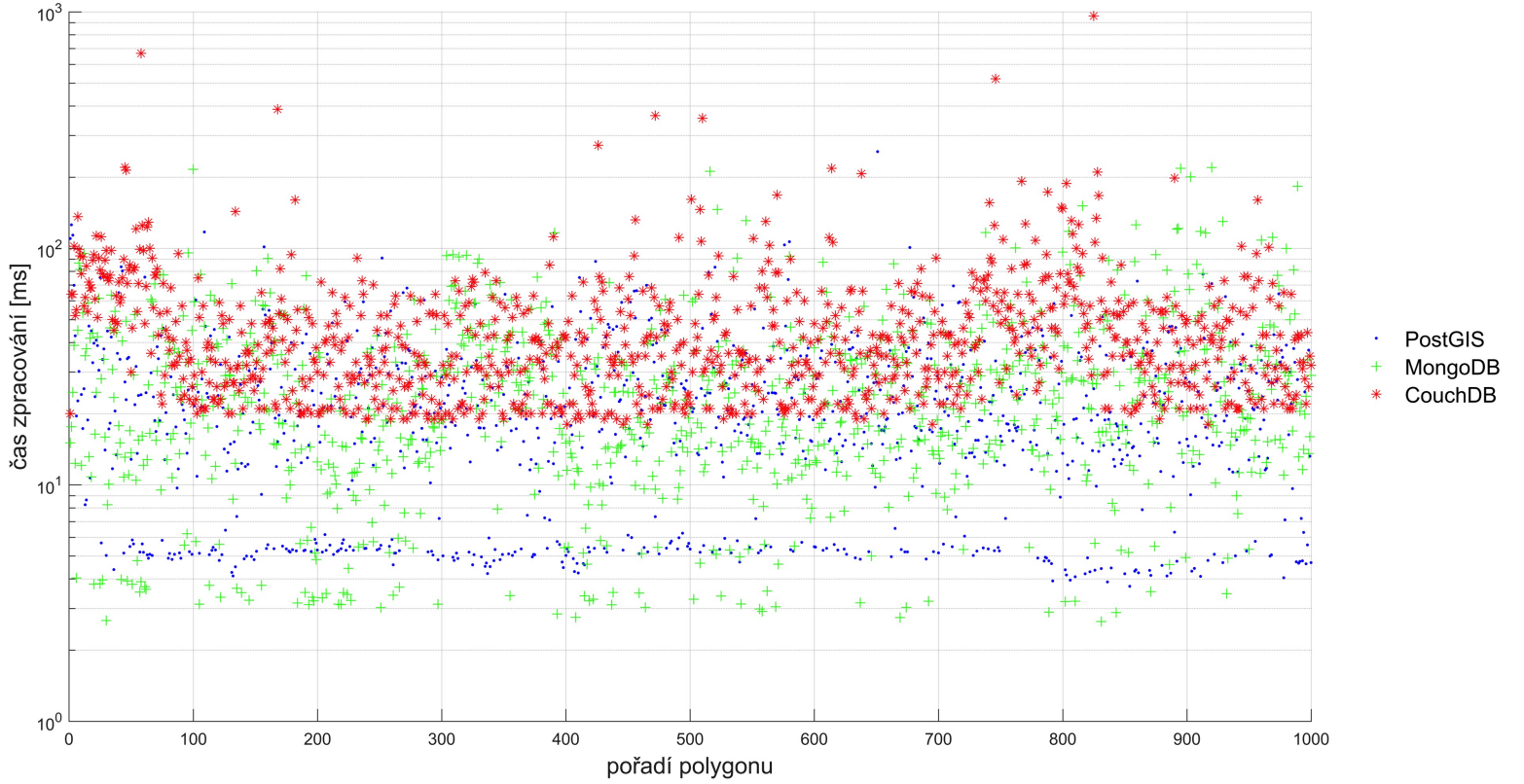
Výsledky testů v grafech



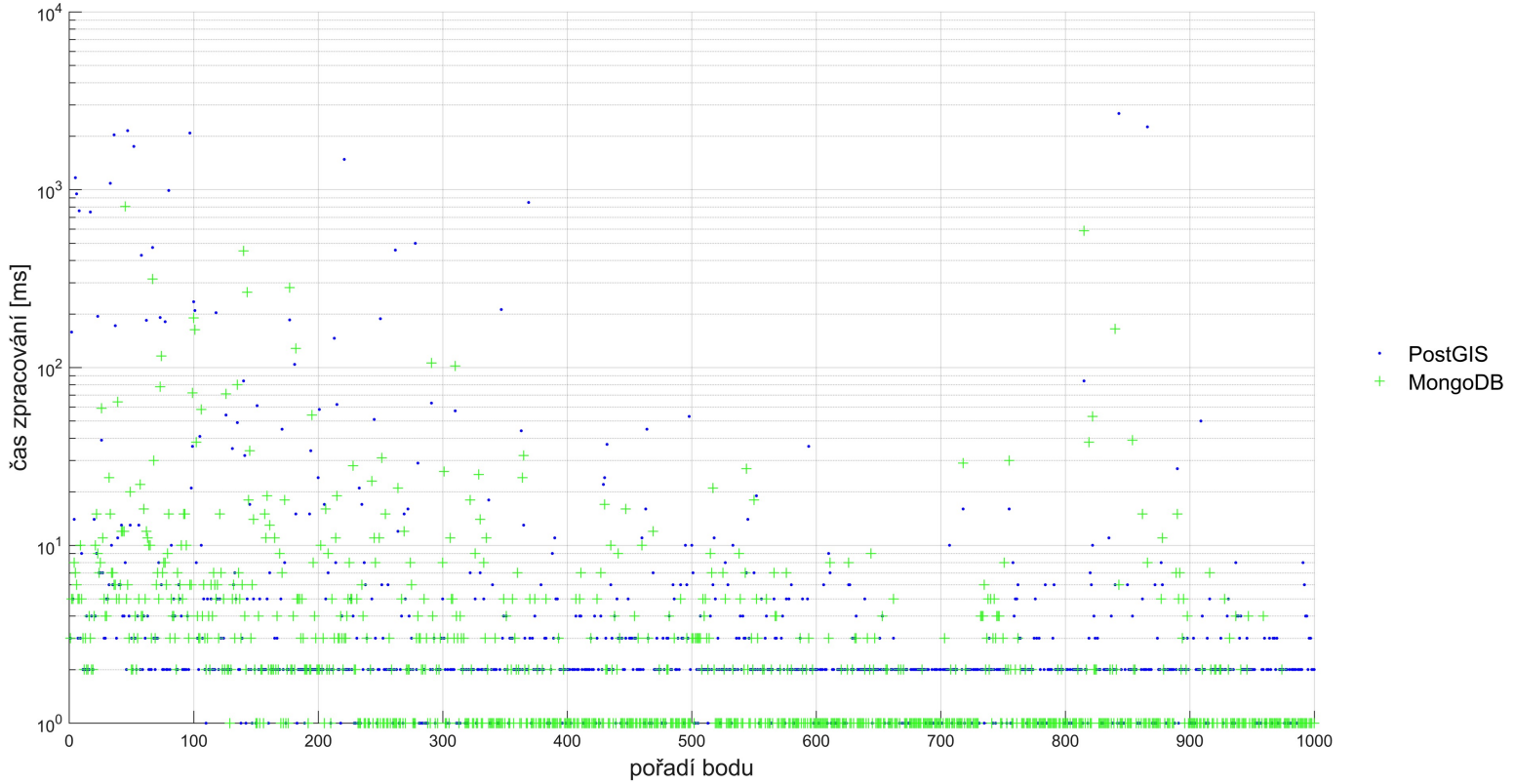
Obrázek B.1: Čas vložení bodů do vybraných SŘBD



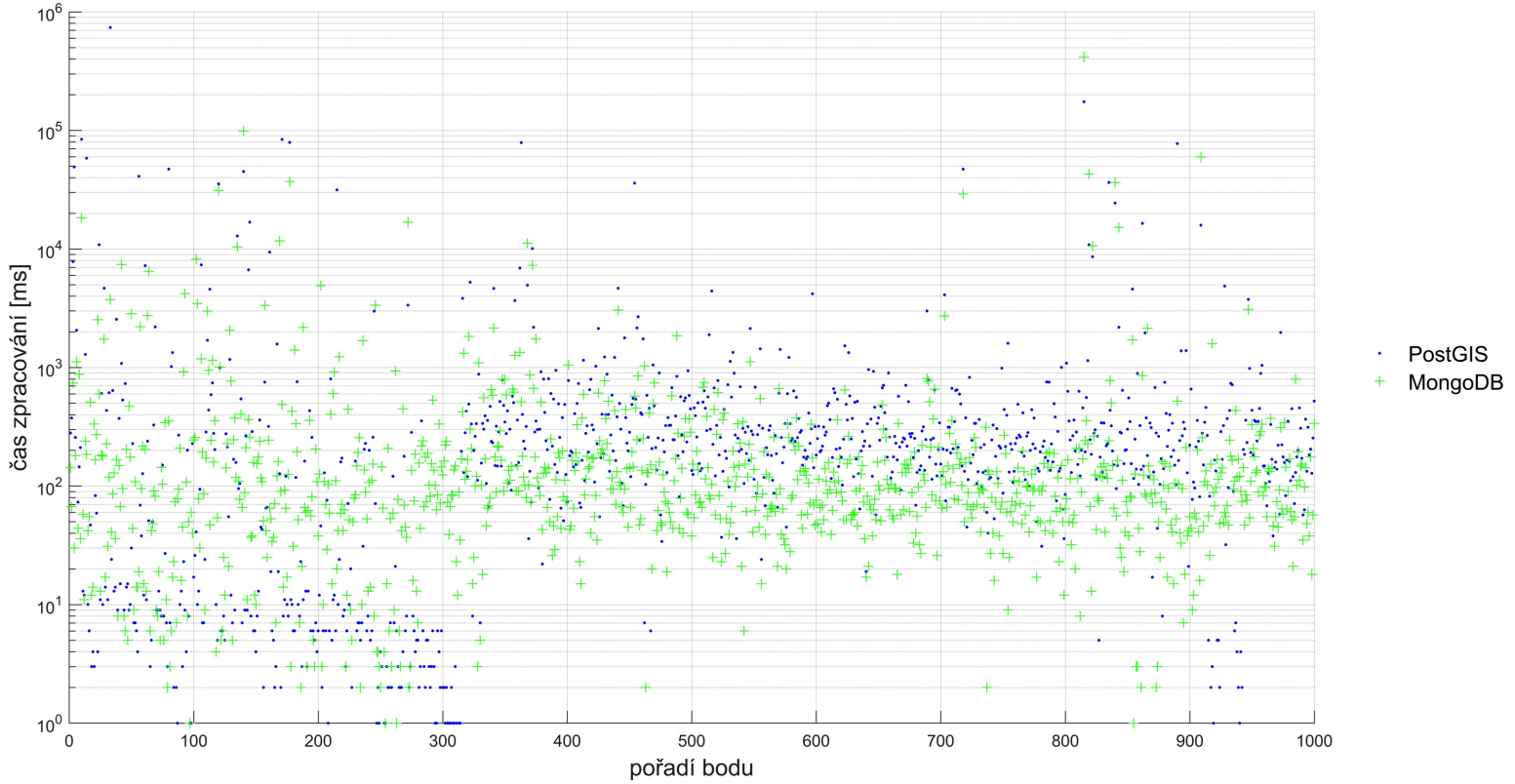
Obrázek B.2: Čas dotazu pro vložení linií do vybraných SŘBD



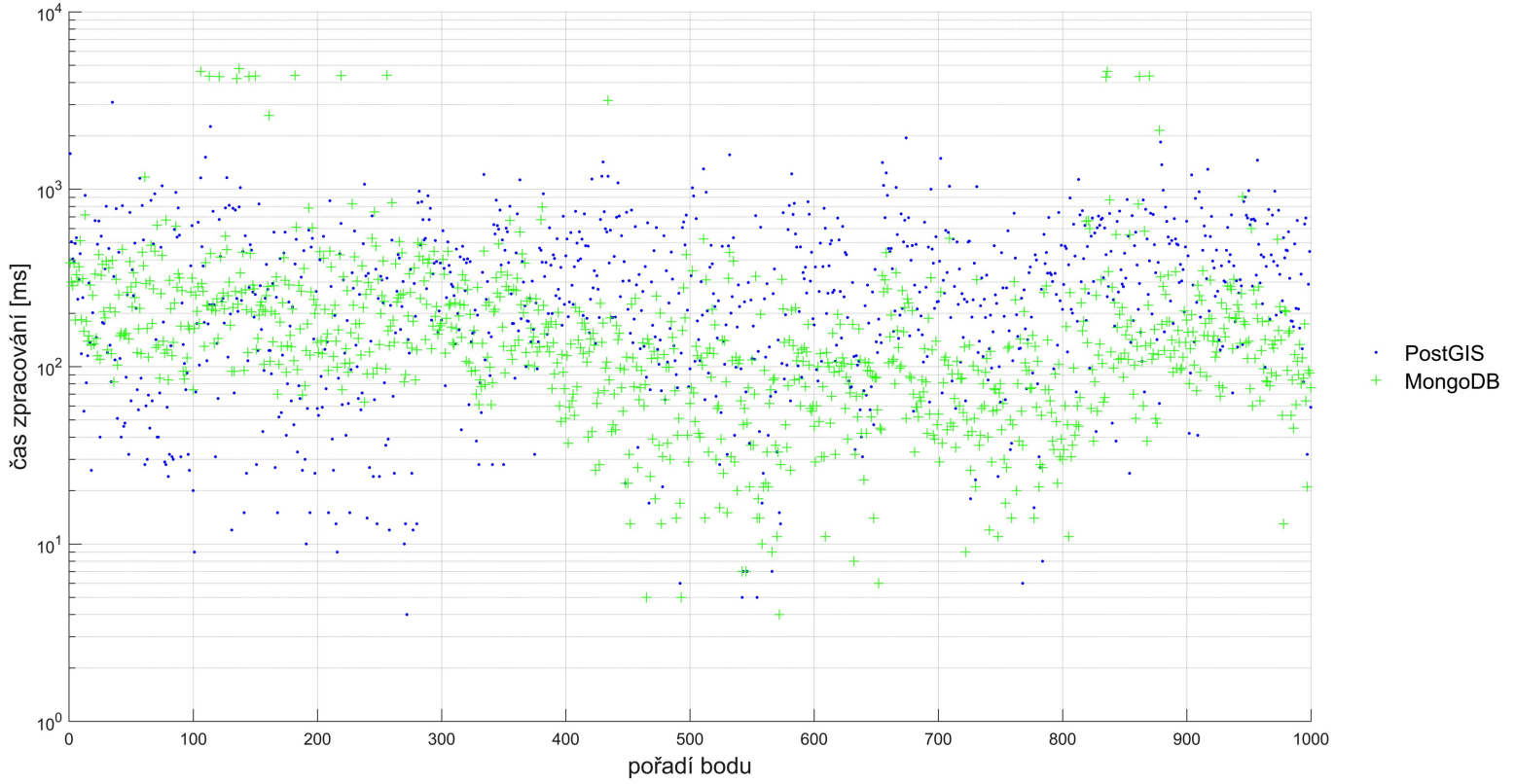
Obrázek B.3: Čas dotazu pro vložení polygonů do vybraných SŘBD



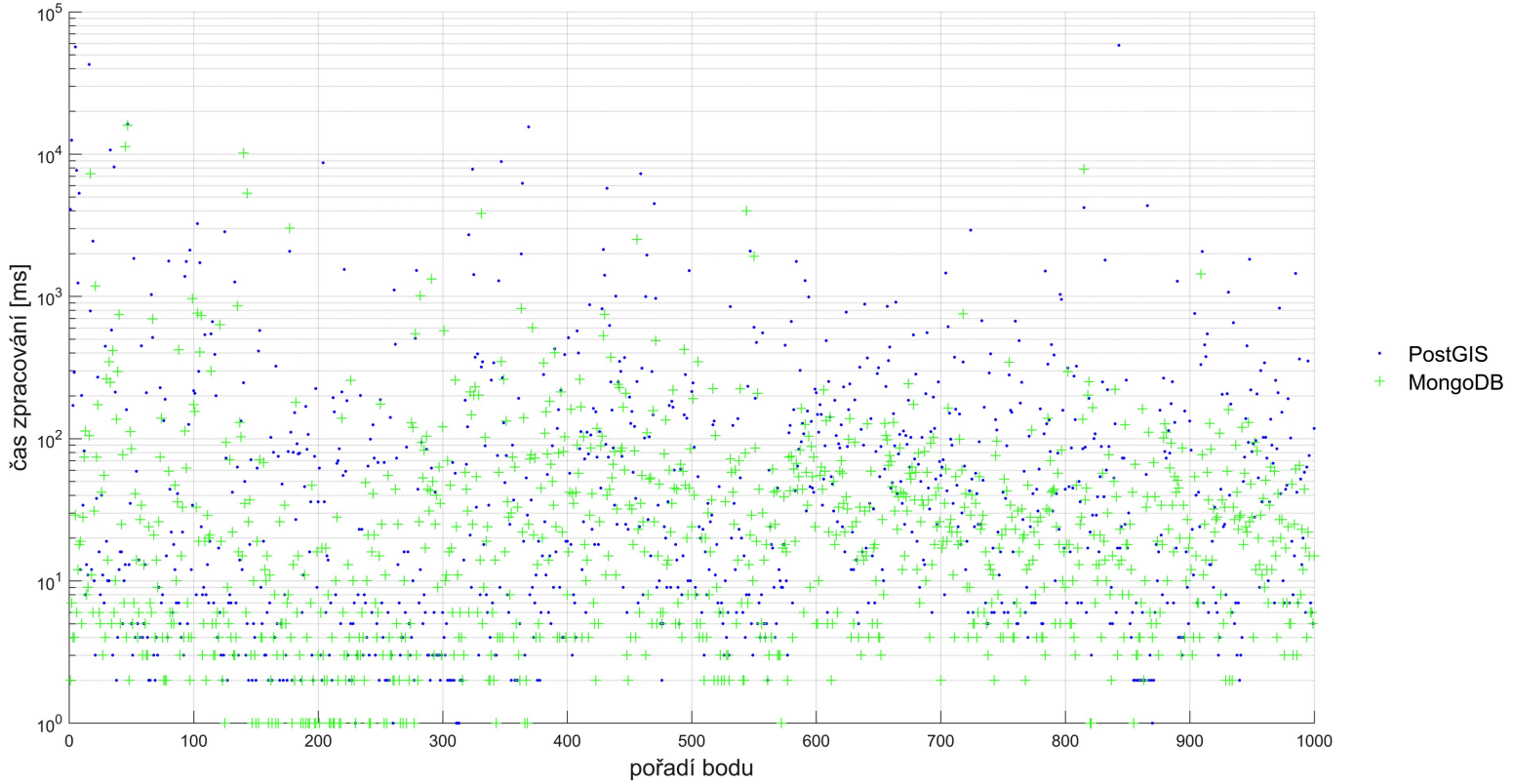
Obrázek B.4: Čas dotazu pro hledání nejbližšího souseda v bodové vrstvě



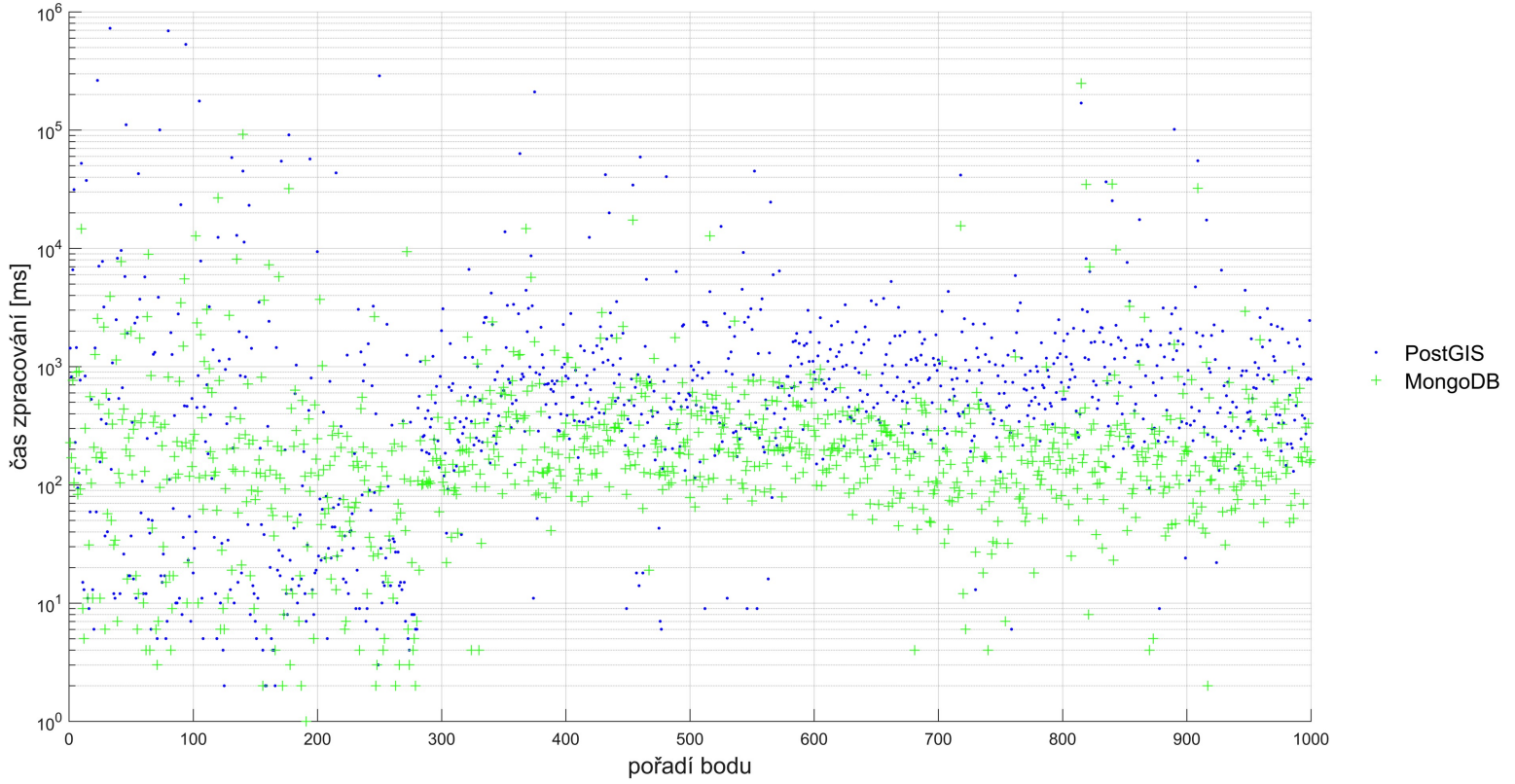
Obrázek B.5: Čas dotazu pro hledání nejbližšího souseda v liniové vrstvě



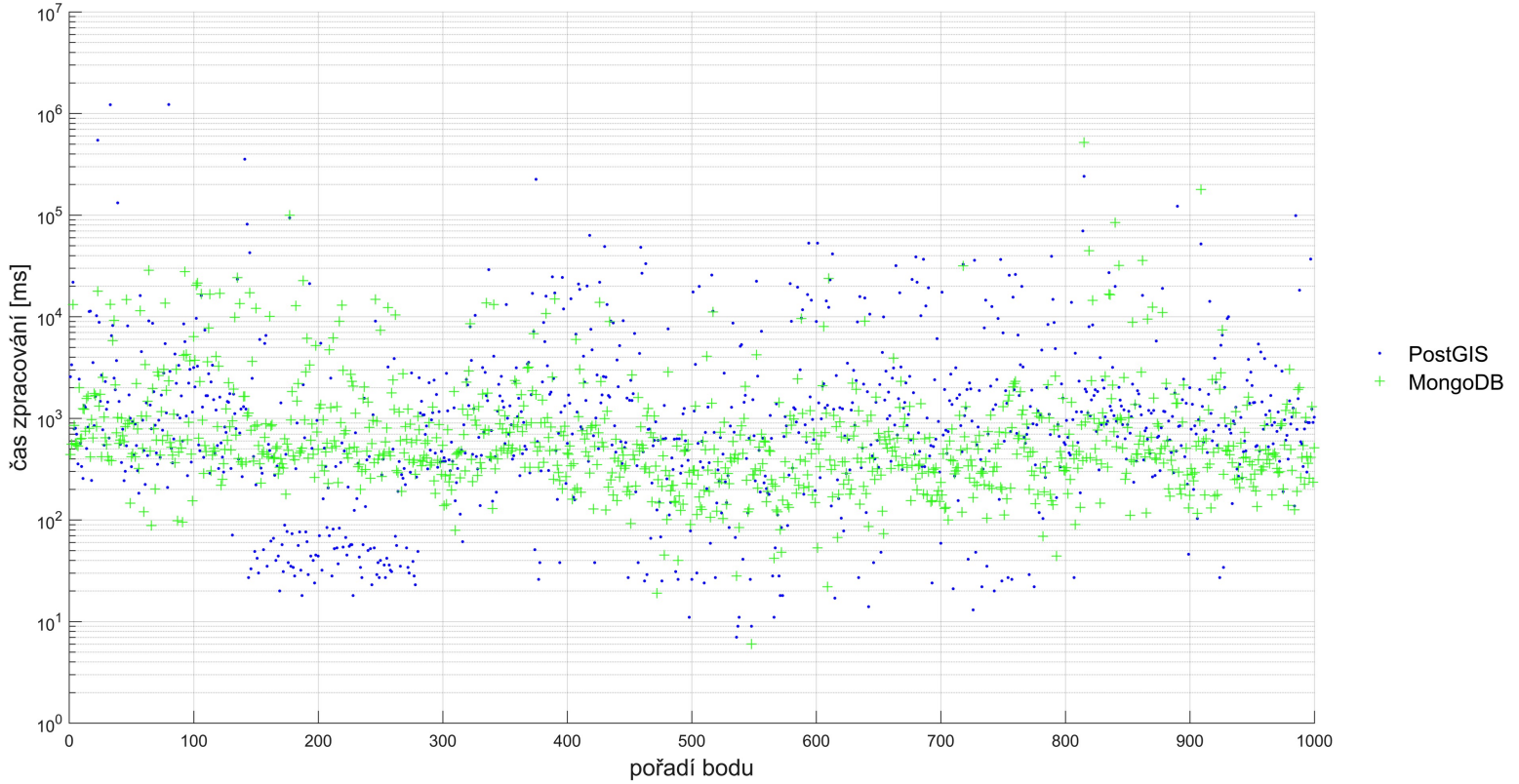
Obrázek B.6: Čas dotazu pro hledání nejbližšího souseda v polygonové vrstvě



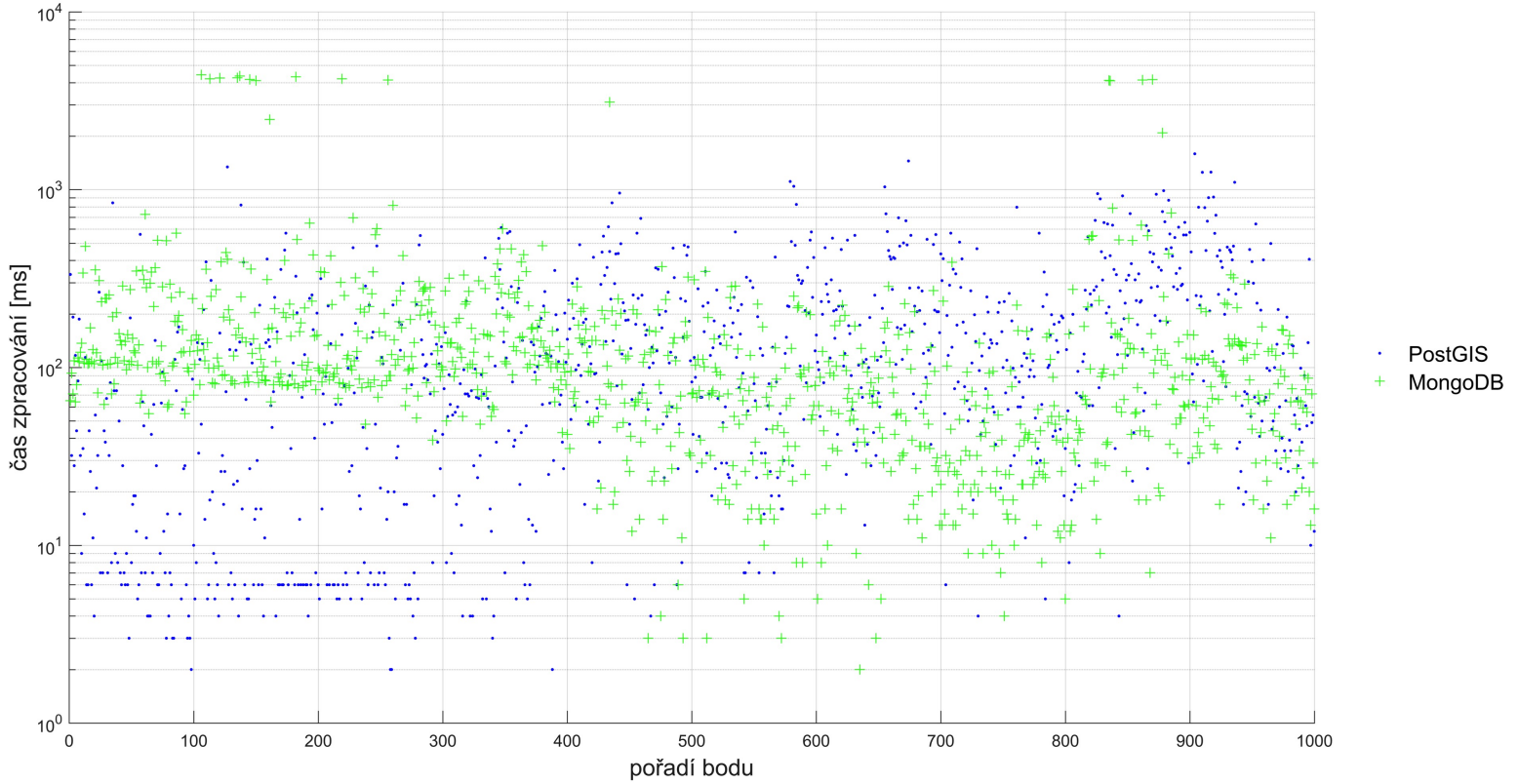
Obrázek B.7: Čas dotazu pro hledání k-nejbližšího souseda v bodové vrstvě



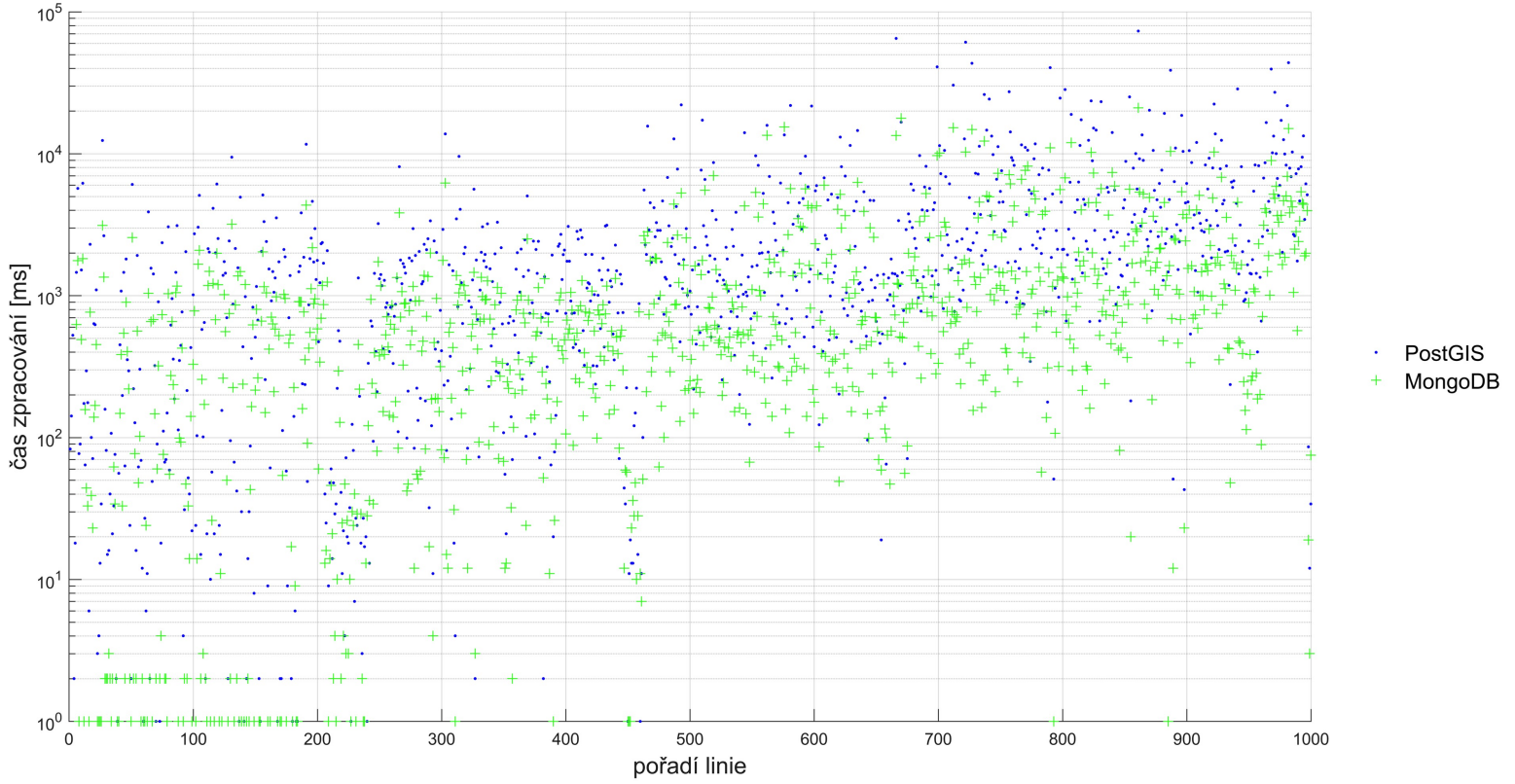
Obrázek B.8: Čas dotazu pro hledání k-nejbližšího souseda v liniové vrstvě



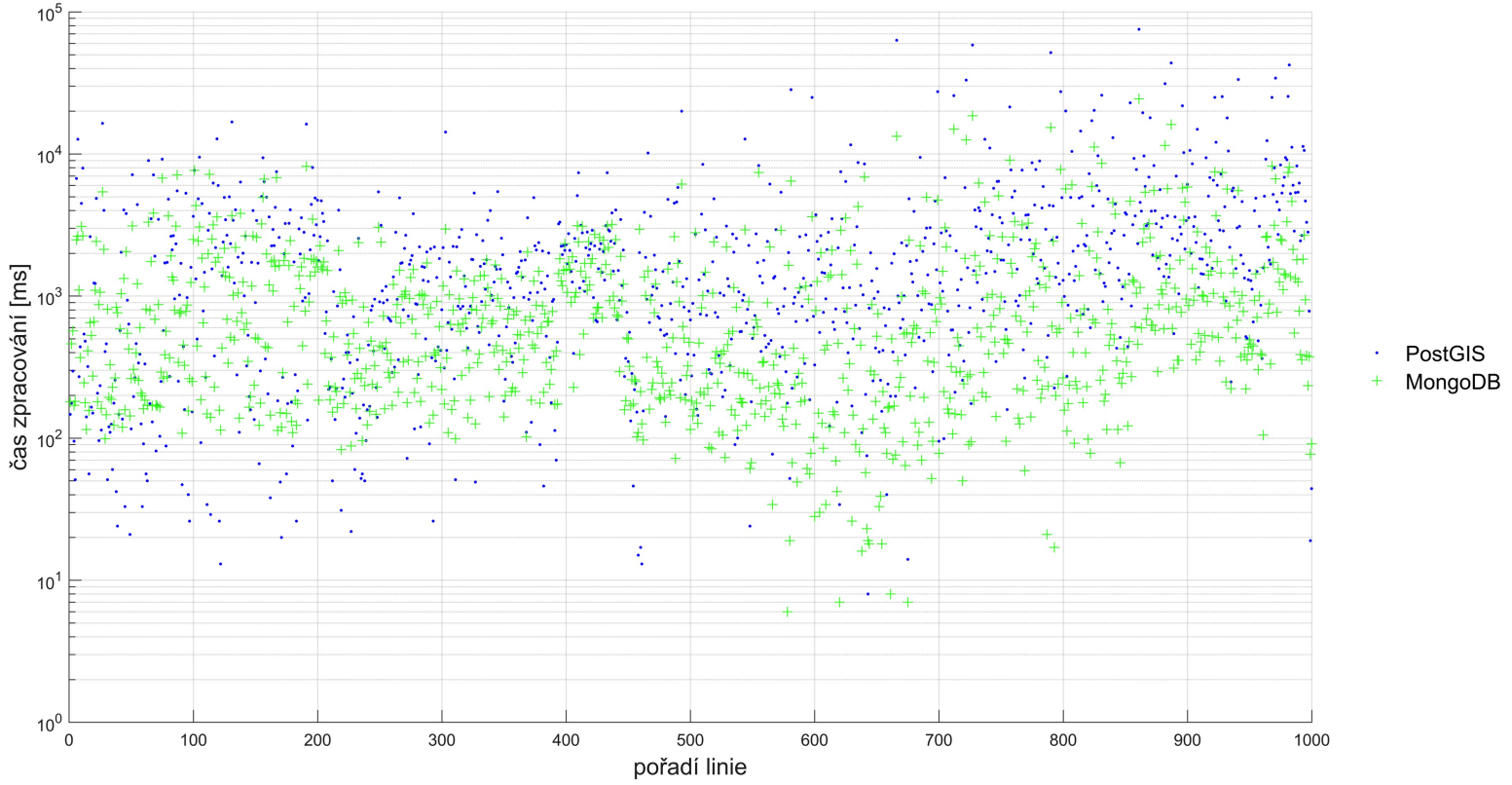
Obrázek B.9: Čas dotazu hledání k-nejbližšího souseda v polygonové vrstvě



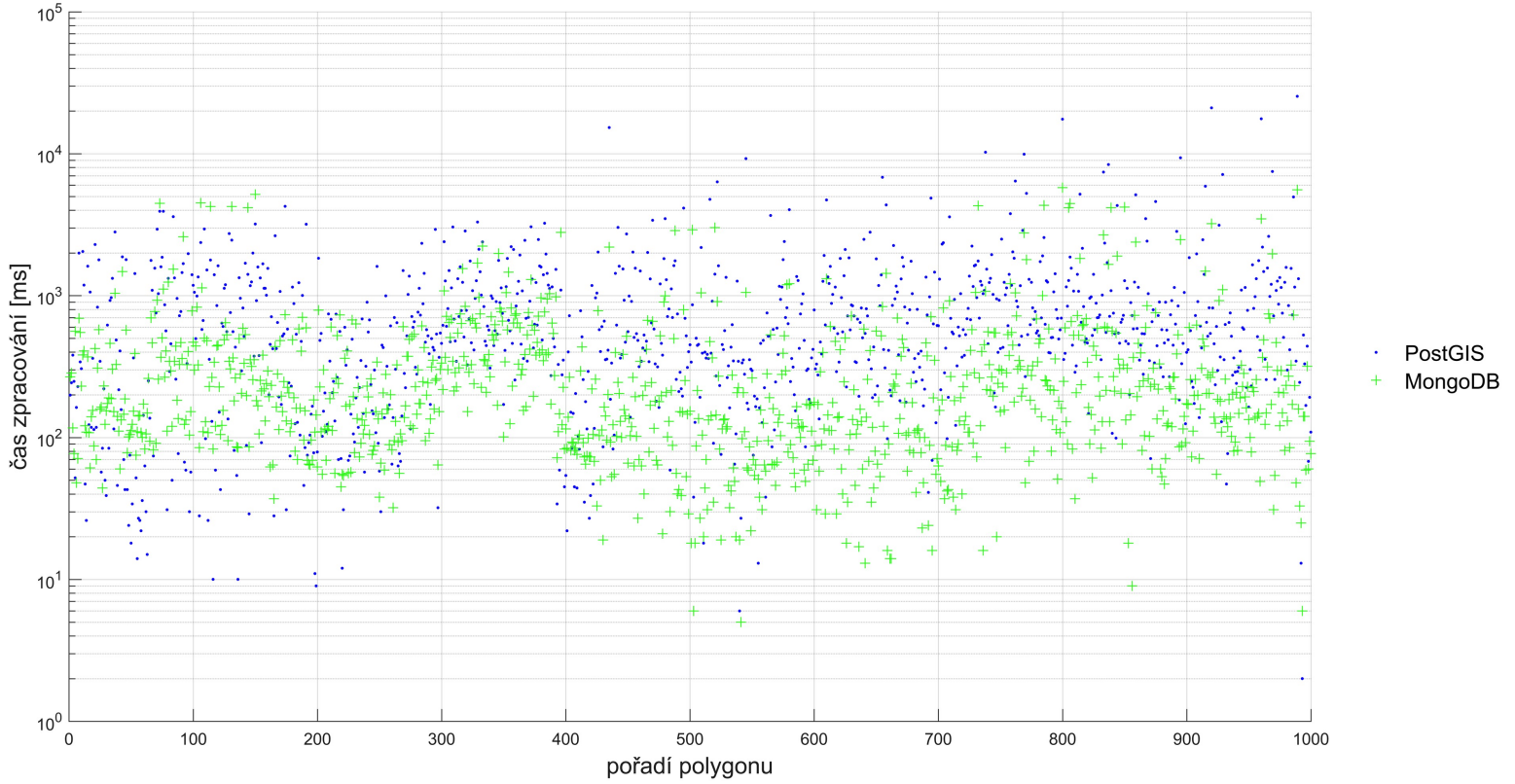
Obrázek B.10: Čas dotazu pro nalezení průniku bodové a polygonové vrstvy



Obrázek B.11: Čas dotazu pro nalezení průniku liniové a linové vrstvy



Obrázek B.12: Čas dotazu pro nalezení průniku liniové a polygonové vrstvy



Obrázek B.13: Čas dotazu pro nalezení průniku polygonové a polygonové vrstvy

Příloha C

Soubor dotazů pro testování

```
01 | INSERT INTO [db].[table]
02 | VALUES (default, ST_Transform (ST_SetSRID (ST_MakePoint
       ( x, y ),4326),3857));
```

Obrázek C.1: Dotaz pro vložení záznamu (bodu), PostGIS

```
01 | db.collection.insert(
02 |         {
03 |                 "type" : "Feature",
04 |                 "geometry" : {
05 |                         "type" : "Point",
06 |                         "coordinates" : [
07 |                          x,
08 |                          y
09 |                          ]
10 |                 },
11 |                 }
12 |         }
13 | )
```

Obrázek C.2: Dotaz pro vložení záznamu (bodu), MongoDB

```
01 |  SELECT id, geometry,
02 |  ST_Distance (ST_Transform (ST_SetSRID (ST_Point ([x,y])
       , 4326), 3857), geometry) as dist
03 |  FROM [db].[table]
04 |  WHERE ST_DWithin (geometry, ST_Transform  (ST_SetSRID (
       ST_Point([x,y]), 4326), 3857), distance)
05 |  ORDER by dist limit 1;
06 |
```

Obrázek C.3: Dotaz pro hledání nejbližšího souseda, PostGIS

```
01 |  db.collection.find(
02 |          {
03 |          geometry: {
04 |                  $near: {
05 |                  $geometry: {
06 |                          type: "Point" ,
07 |                          coordinates: [ x, y ]
08 |                          },
09 |                          $maxDistance: <distance>,
10 |                  }
11 |                  }
12 |          }
13 |   ).limit(1)
```

Obrázek C.4: Dotaz pro hledání nejbližšího souseda, MongoDB

```
01 |  SELECT id, geometry,
02 |  ST_Distance (ST_Transform (ST_SetSRID (ST_Point ([x,y])
       , 4326), 3857), geometry) as dist
03 |  FROM [db].[table]
04 |  WHERE ST_DWithin (geometry, ST_Transform  (ST_SetSRID (
       ST_Point([x,y]), 4326), 3857), distance)
05 |  ORDER by dist limit k;
06 |
```

Obrázek C.5: Dotaz pro hledání k-nejbližšího souseda, PostGIS

```
db.collection.find(
        {
        geometry: {
                $near: {
                $geometry: {
                        type: "Point" ,
                        coordinates: [ x, y ]
                        },
                        $maxDistance: <distance>,
                }
                }
}
).limit(k)
```

Obrázek C.6: Dotaz pro hledání k-nejbližšího souseda, MongoDB

```
SELECT id
FROM [db].[table]
WHERE ST_Intersects (geometry, ST_Transform (ST_SetSRID
    (ST_GeomFromText ('POINT ( x y )'), 4326), 3857));
```

Obrázek C.7: Dotaz pro hledání průniku, PostGIS

```
db.collection.find(
        {
        geometry: {
                $geoIntersects: {
                $geometry: {
                        type: "Point" ,
                        coordinates: [ x, y ]
                        },
                }
        }
}
)
```

Obrázek C.8: Dotaz pro hledání průniku, MongoDB

```
01 | SELECT a.id as aliasA, b.id as aliasB
02 | FROM [db].[table] as a, [db].[table] as b
03 | WHERE ST_intersects(a.geometry, b.geometry);
```

Obrázek C.9: Dotaz pro prostorové spojení, PostGIS