Ukládání rozsáhlých dat v NoSQL databázích

František Koleček, xkolec08@stud.fit.vut.cz
Tomáš Moravčík, xmorav41@stud.fit.vut.cz
David Sladký, xsladk07@stud.fit.vut.cz

zima 2022

Obsah

I Analýza zdrojových dat a návrh jejich uložení v NoSQL databázi **1**

1 Analýza zdrojových dat **2**

2 Návrh způsobu uložení dat **3**

3 Zvolená NoSQL databáze **4**

II Návrh, implemetace a použití aplikace **5**

4 Návrh aplikace **6**
4.1 Zpracování souborů XML 6
4.1.1 Struktura XML souborů 6
4.1.2 Zvolené Prostředky 6
4.1.3 Způsob implementace 7

5 Způsob použití **8**

6 Experimenty **9**
6.1 Rychlost zpracování XML souborů 9

Část I

Analýza zdrojových dat a návrh jejich uložení v NoSQL databázi

Kapitola 1

Analýza zdrojových dat

Použitá datová sada se nachází na stránkách ministerstva dopravy a to konrétně zde. Detailní popis formátu dokumentů datové sady lze najít na stejném místě.

Datová sada se skládá z `XML` souborů. Tyto soubory jsou zveřejňovaný na začátku roku pro celý rok ve složce `GVD`. Dále je pak každý měsíc zveřejňována datová sada pro daný měsíc s aktualizacemi pro spoje. Každý soubor má element `CZPTTCreation`, který určuje jeho vytvoření.

`XML` soubory lze rozdělit do následujících tří skupin:

- Definující spoj
- Rušící spoj
- Definující náhradní spoj

Soubory definující spoje mají jako kořenový element `CZPTTCISMEssage`. První důležité informace se nachází v elementu `Identifiers`, kde jsou uvedeny identifikátory pro definované spojení a vlak, který ho bude provádět. Dále element `CZPTTHeader` určuje, zda spoj přijíždí nebo pokračuje z/do zahraniční stanice. Elementy `CZPTTLocation` obsahují jednotlivé stanice, kterýma vlak projíždí. Zde jsou uvedeny i další informace ke stanici. Nejvýznamnější z nich jsou: čas příjezdu/odjezdu, typ aktivity. Po uvedení všech stanic následuje element `PlannedCalendar`, který určuje výčet dní, kdy je tento spoj prováděň.

Soubory rušící spoje mají kořenový element `CZCanceledPTTMessage`. Podobně jako soubory definující spoje obsahují identifikaci spoje, který se ruší a výčet dní, kdy se ruší. Dále už nenesou žádné informace.

Soubory definující náhradní spoje mají stejnou strukturu jako soubory definující spoje jenom s jediným rozdíle. Obsahují element `RelatedPlannedTransportIdentifiers`, který určuje, jaký spoj nahrazují. Tyto spoje mají unikátní indentifikátor vůči normálním spojům.

Kapitola 2

Návrh způsobu uložení dat

Cíl: Po posouzení vlastností dat (z předchozí analýzy) a očekávaných dotazů (ze zadání) navrhnout vhodný způsob uložení dat do NoSQL databáze. Způsob uložená musí být vhodný z hlediska způsobu nahrávání dat ze zdroje do databáze (a to i průběžného doplňování či aktualizace, bez smazání celé databáze) a z hlediska rychlosti dotazování dat v databázi z aplikace s využitím vlastností NoSQL (s využitím klíčů a škálovatelnosti/distribuovanosti databáze). Data lze při nahrávání ze zdroje do databáze předzpracovávat, např. kombinovat či doplňovat, odvozovat pomocná data, předpočítávat agregace, atp. Takové předzpracování může trvat déle (kritérium vhodnosti při předzpracování v průběhu nahrávání není čas, ale vhodné využití obecných vlastností NoSQL, jako je sharding).

Obsah: Pro skupinu či každou podstatnou vlastnost dat z analýzy a dotaz ze zadání (pokud bude mít vliv na návrh) popsat, co znamená, jaký problém představuje, jaké je řešení, proč je zvolené řešení dobré a stručně jaké jsou případné alternativy.

Prostředky: Strukturovaný text (odstavce, sekce, odrážky, atd.), kde je popsán proces získání, předzpracování a uložení dat ze zdroje do databáze. Možno použít také pseudokód či diagramy popisující datové toky a použité struktury a vlastnosti NoSQL databází obecně. Každé návrhové rozhodnutí musí být řádně zdůvodněno (např. části se strukturou "dotaz/vlastnost", "problém", "řešení", "důvod", "alternativy").

Fáze projektu: Po analýze dat a analýze uživatelských požadavků na aplikaci, většinou souběžně s návrhem aplikace.

Kapitola 3

Zvolená NoSQL databáze

Cíl: Rozhrnout a zdůvodnit jaký druh NoSQL databáze je vhodný (zdůvodnění plyne částečně již z předchozího návrhu) a jaký konkrétní produkt NoSQL databáze bude použit.

Obsah: Určit typ databáze a konkrétní produkt NoSQL, vypsat jeho vlastnosti, které jsou pro toto řešení užitečné (a jiné než u jiných typů a produktů NoSQL) a zdůvodnit jejich vhodnost v kontextu předchozího návrhu.

Prostředky: Stručný volný text (až několik kratších odstavců) s případným vyznačením podstatných části.

Fáze projektu: Zakončování návrhu a přechod k implementaci.

Část II

Návrh, implemetace a použití aplikace

Kapitola 4

Návrh aplikace

4.1 Zpracování souborů XML

Informace o vlakových spojích jsou získávány ze souborů ve formátu XML. Tyto soubory je třeba zpracovat – nahrát data do strukturované vnitřní reprezentace programu, pro efektivní nahrávání do databáze. Každý soubor obsahuje informace o jednom konkrétním vlakovém spoji, jeho cesta a časy ve stanicích jsou vždy stejné, jsou zde definované dny, ve kterých tento spoj jede. Nejdůležitější zpracovávaná data jsou:

- Identifikátory
- Navštívené stanice a časy příjezdu a odjezdu
- Dny, ve kterých spoj jede

4.1.1 Struktura XML souborů

Každá s těchto informací se nachází ve vlastní „větvi“ v souboru. Nachází se u nich samozřejmě i dodatečné informace – detaily lokace, činnost vlaku ve stanici atd. Platné dny jsou určeny pomocí dvou atributů – začátek a konec platnosti a bitmapa. Bitmapou je myšlen řetězec jedniček a nul, kde jedničky vyjadřují platnost v jednotlivých dnech. Tímto způsobem lze vyjádřit platné dny na rok dopředu pomocí řetězce dlouhém 365 znaků.

4.1.2 Zvolené Prostředky

Zpracovávání souborů je implementováno v jazyce Python v souboru `parser.py`. Je využito modulu `xml`, který je součástí základní instalace Pythonu, není třeba jej dodatečně instalovat. Tento modul obsahuje třídu `ElementTree`, která umožňuje snadné nahrávání dat ze stromové struktury souboru. Velmi užitečnou funkcí je možnost adresování uzlů pomocí „cesty“ - obdobným způsobem jako adresování souborů v souborovém systému.

4.1.3 Způsob implementace

Nejdůležitější roli při zpracovávání hraje funkce `node_to_dict`, která rekurzivně prohledává daný uzel a převádí jeho obsah na slovníkový datový typ.

Protože platné dny jsou v naší databázi ukládány jako vlastní uzly, na které pak odkazují spoje, které jsou platné v daný den, je potřeba původní reprezentaci platných dní (popsána výše) převést na seznam konkrétních kalendářních dat. Pro implementaci této funkcionality byl využit modul `datetime`, který je opět součástí základní instalace pythonu. Tento modul mimo jiné umožňuje vykonávat „aritmetické operace“ s kalendářními daty. V našem případě se jedná o sečtení data začátku platnosti a indexu jedničky v příslušné bitmapě. Implementace je ve funkci `cal_to_listofdays`.

Kapitola 5

Způsob použití

Cíl: Poskytnout stručnou dokumentaci pro zprovoznění databáze a aplikace.

Obsah: Stručně popsat, jak celé řešení zprovoznit, tj. nasadit databázi i aplikaci vč. způsobu volání aplikace (příkazový řádek, parametry) pro úlohy předzpracování a nahrání dat ze zdroje do databáze a pro ulohy hledání nad databází tak, jak byly definovány v zadání.

Prostředky: Stručný text obsahující návod (popis) s ukázkami způsobu volání aplikace (např. pro skripty by to byl kód příkazového řadku).

Fáze projektu: Dokončování implementace, chystání dokumentace pro předání výsledného systému zákazníkovi.

Kapitola 6

Experimenty

Cíl: Změřit, jak aplikace a databáze fungují v praxi.

Obsah: Popis výchozí konfigurace aplikace a nasazení databáze stroje, kde budou experimenty probíhat (HW a SW). Popis experimentů typicky představující nahrání dat ze zdroje do databáze či dotazy ze zedání s výslednými časy jejich provedení. Případné poznámky k výsledkům experimentů.

Prostředky: Strukturvaný text, případně tabulka či graf s doprovodným textem.

Fáze projektu: Testování řešení před předáním výsledného systému zákazníkovi.

6.1 Rychlost zpracování XML souborů

Funkčnost a efektivita skriptu `parser.py` byla testována nad složkou `GVD2022`, která obsahuje přibližně 12300 souborů formátu XML. Všechny soubory byly v experimentu zpracovány a výpis důležitých informací vytisknut na standardní výstup. Tato operace trvala 93 sekund, tedy cca 132 zpracovaných souborů za sekundu.