ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ

FAKULTA STAVEBNÍ

OBOR GEOMATIKA

Semestrální práce

UZPD - Historické vlny

Vedoucí práce: Ing. Martin Landa, Ph.D.

Únor 2018

Lubomír Bucek, Zuzana Richtrová

Obsah

Úvod

Cílem projektu, který tvoří semestrální práci předmětu Úvod do zpracování prostorových dat bylo vytvořit databázi naplněnou volně dostupnými daty. Jelikož téma bylo takto zadáno velmi volně, po konzultaci bylo zvoleno téma původně zadáno společností Vesseltracker. Zadání bylo specifikováno jako vytvoření automatizovaného skriptu, který v pravidelných intervalech stahuje historická data o výšce vln pro všechny oceány a po jejich zpracování je uloží do NoSQL databázového systému Elasticsearch.

Vesseltracker svým software a službami poskytuje vylepšené výstupy ze systémů celosvětového sledování pohybu lodí - AIS. Systém automatické identifikace lodí (AIS) se skládá ze tří sektorů, uživatelský (přijímače na lodích), satelitní (multifunkční satelity, jejichž jednou z funkcí je dekódování a přeposílání zpráv z lodních AIS) a nakonec pozemní (antény podobné těm satelitním, umístěné na pobřeží obvykle s dosahem okolo 300km). Co se právního zakotvení týče, každá loď navržená k převozu většího množství lidí musí mít zařízení AIS. Pro nákladní lodě platí pravidlo, pokud má loď větší výtlak než 300 tun, tak musí mít přijímač také.

Vesseltracker jako jeden ze svých produktů nabízí i webovou mapovou aplikaci, která umožňuje živé sledování veškerého pohybu po světových mořích a řekách, historická data o pohybu jednotlivých lodí, sledování vlastních skupin lodí a další placené nebo neplacené služby. V současné době nabízí Vesseltracker jako doplněk do webové mapové aplikace předpověď počasí a vln na příštích 24 hodin formou překryvových polygonů oblastí s nepříznivými podmínkami, což lze využít pro plánování cest například v prostoru bouří. Co ovšem v portfoliu chybělo bylo poskytování historických informací o počasí nebo vlnách v daném místě na zemi v libovolném čase. Nejedná se ani tak o zbytečný výmysl, jak by se mohlo zdát. Jeden z případů potenciálního využití bylo například neplánované zmrznutí převáženého nákladu na nákladní lodi v oblasti okolo břehů Kanady. Otázkou bylo, zda pochybil personál obsluhy lodi - chybné nastavení regulace teploty nákladu anebo za vše mohly nepříznivé a nečekané klimatické podmínky v oblasti. Například tento konkrétní požadavek vedl k žádosti o vývoj služby pro poskytování historických dat o počasí a vlnách.

Zdrojová data

Požadavkem projektu bylo plošné pokrytí celé země - konkrétně tedy pouze oceánů, ale rozhodně nešlo použít pouze bodová data z bójí a to v co nejlepším možném rozlišení. Po prozkoumání dostupných datových zdrojů došlo ke zvolení celosvětových modelů produkovaných NOAA National Weather Service - konkrétně částí National Centers for Environmental Prediction. NCEP poskytuje národní (USA) a celosvětové modely počasí, vodstva a klimatu, předpovědi, varování a analýzy, týkající se výše zmíněného.

Použitý model pro výšku vln se jmenuje Wavewatch III, konkrétně jeho “multi-grid wave model”. Model je počítán čtyřikrát denně (00,06,12,18) a produkuje 9, 6 a 3 hodiny předpovědi do minulosti a celkem 180 hodin předpovědi do budoucnosti po tříhodinových intervalech. Data jsou publikována jako rastrové soubory typu GRIB-2 (Gridded binary). O jeho standardizaci se stará World Meteorological Organization's Commission for Basic Systems, konkrétní implementace je uvedena pod pořadovým číslem GRIB FM 92-IX a je popsána ve WMO Manual of Codes 306. Data jsou v rozlišení 0,5° x 0,5° pro celou zeměkouli - pevniny jsou v modelu obsaženy také, ovšem pochopitelně nemají žádné hodnoty. Samotné detaily modelu, který leží v pozadí generovaných dat, zde nebudou vyloženy, zájemce si je může vyhledat. Pro potřeby projektu jsou používány dva typy zdrojových dat:

1) Tzv. production data, uchovávána po dobu sedm dní zpět a obsahujících předpověď na až 180 hodin dopředu, která lze čerpat například pomocí FTP protokolu ze stránky ftp://ftpprd.ncep.noaa.gov/pub/data/nccf/com/wave/prod/ Tato data jsou publikována každé tři hodiny. Ukázkový název souboru ze složky /multi_1.20180130: multi_1.glo_30mext.t06z.f003.grib2 - obsahuje globální data se všemi parametry, které jsou v souboru uvedeny (je jich celkem 16), z 9:00 (soubor t06, ovšem předpověď na od nyní za +3 hodiny. Jejich velikost je vždy 2,7MB.
2) Historická data obsahující v podstatě stejná data jako production, ovšem ve formátu 1 soubor na celý měsíc. Tato data nejsou publikována nijak pravidelně a může se stát, že například po dobu tří měsíců nejsou tato data vůbec ke stažení. K prohlížení dostupných dat i jejich stažení lze využít například odkazu na webovou stránku http://polar.ncep.noaa.gov/pub/history/waves/multi_1 Ukázkový název souboru: multi_1.glo_30m.hs.201712.grb2 - obsahuje globální data o výšce vln (hs) v rozlišení 30 stupňů z prosince 2017. Jejich velikost se pohybuje okolo 27MB (dle počtu dní v měsíci).

Popis a rozbor problému

Jelikož se v případě GRIB2 jedná o v běžném životě poměrně nerozšířený datový formát zdrojových dat, který se ovšem společně s NetCDF stal standardem pro výstup z klimatických a jiných modelů, bylo pro jeho zpracování vytvořeno nezanedbatelné množství nástrojů, které ale obvykle pracují v prostředí příkazové řádky. V tomto projektu byl použit nástroj ecCodes, vytvořený European Centre for Medium-Range Weather Forecasts a publikovaný pod licencí Apache Licence 2.0. Poskytuje po zkompilování nástroje pro práci se soubory WMO FM-92 GRIB edition 1 a edition 2 a také WMO FM-94 BUFR edition 3 a edition 4. Nedávno bylo též přidáno i nízkoúrovňové Python rozhraní, ale to nebylo po nepovedených pokusech nakonec použito. Jak již bylo řečeno, jedná se o nástroj pouštěný z příkazové řádky. Z balíku software ecCodes ovšem byly využity pouze dva nástroje a to:

grib_ls - Vypisuje obsah GRIB souboru vytištěním hodnot zadaných klíčů do standardního výstupu. Pomocí parametru -l například specifikujeme hledané místo nebo -f charakterizuje formát výstupních dat, případně -w key=value limituje zobrazované klíče (v našem případě např. shortName = swh - significant wave height). Tento příkaz byl použit pouze pro prozkoumání datové sady a testování korektnosti běhu programu.

grib_get_data - Vypisuje hodnotu daného klíče pro všechny hodnoty kombinací zem. šířky a zem. délky do standardního výstupu. Tento příkaz byl tedy použit pro samotné vytažení “užitečných dat” z GRIB formátu a zpracování v dalších fázích.

Pro samotné řízení běhu programu ovšem byly vytvořeny Python skripty, protože se to ukázalo jako nejpohodlnější řešení. Jejich funkce budou popsány v kapitole Dokumentace programu. Jelikož celé řešení mělo být navrženo jako bezúdržbové (nejspíš naivní sen, ale což), není možné, aby Python skripty byly spouštěny ručně každý den člověkem manuálně. Z toho vzešla potřeba použít Linuxový program Cron, který úpravou souboru crontab umožňuje spouštět v předem nastavené doby dne přichystané příkazy. Ačkoliv by nyní stačilo spouštět Python skripty, bylo celé celé prostředí navrženo, aby pro větší kontrolu a logování akcí docházelo ke Pythonu pomocí Bash skriptů. Ačkoliv se to může zdát krkolomné, ukázalo se to jako velmi elegantní řešení.

Elasticsearch

Při výpočtu objemu dat zjistíme, že se jedná o potenciálně obrovské množství (720 zem. délek * 360 zem. šířek * 8 krát za den * 30 dní v měsíci) s byť jen třemi atributy atributy (výška vlny, čas, poloha) dává tento přístup měsíčně 62 milionů záznamů, každý se třemi atributy. Tento objem dat vyžadoval zvážení použití No-SQL databáze pro alespoň rozumnou dobu odezvy vyhledávaných dat. Jelikož má Vesseltracker již delší dobu výborné zkušenosti se systémem Elasticsearch, nakonec se rozhodlo použít jej i pro tento projekt.

Jedná se o vyhledávací a úložný systém založený na Apache Lucene. Disponuje RESTful rozhraním a nabízí vysokou dostupnost, rychlost a škálovatelnost. Je vyvíjený v Javě a je v jedné ze svých variant šířen zdarma pod licencí Apache, zároveň jsou dostupné i placené firemní varianty, kde je Elasticsearch společně s dalšími vizualizačními, monitorovacími a bezpečnostními programy a službami. Jedná se o distribuovaný systém. Pokud výkon serveru nestačí, stačí přidat další server. Takto vzniklý cluster pak rozloží data optimálně mezi vzniklé uzly. Fulltextové vyhledávání umožňuje podporu více jazyků, vyhledávání na základě geografické polohy (využito v případě vln), vyhledávání podobných nebo příbuzných záznamů apod. Elastic využívá API - téměř každá akce může být provedena pomocí JSON dokumentu, který je zasílán přes HTTP. Pro mnoho programovacích jazyků také existují knihovny zjednodušující práci s Elasticsearch - samozřejmě i pro Python, který byl pro posílání dat do Elastic využit. Pro potřeby projektu nebylo třeba seznamovat se s administrací ES clusterů, jelikož to si samozřejmě Vesseltracker zařídil sám. Elasticsearch je bezschémovou databází. Je ovšem nutné namapování proměnných před samotným vkládáním.

Níže je vidět namapování proměnných pro zmíněný projekt.

```
"mappings": {
    "waves": {
      "properties": {
        "swh": {
          "index": "no",
          "type": "double"
        },
        "location": {
          "type": "geo_point"
        },
        "ts": {
          "type": "date"
        },}}
```

Je tedy vidět že při vkládání dochází k indexování polohy a času (ts) pro zrychlené vyhledávání, ovšem výšky vln ne.

Ukázka vyhledávacího dotazu (query):

```
  index: 'waves_*',
  body: {
    size: 10000,
    query: {
      "bool": {
        "must": [{
            "range": {
              "ts": {
                "gte": 1512110100,
                "lt": 1512444900
              }}}],
        "filter": {
```

```
          "geo_distance": {
             "distance": "55km",
             "location": {
                "lat": 53.75,
                "lon": 5.2
             }}}}},
sort: [{
      "_geo_distance" : {
         "location": {
            "lat": 53.75,
            "lon": 5.2
         },
         "order" : "asc",
         "unit" : "km",
         "mode" : "min",
         "distance_type" : "sloppy_arc"
      }},
   {
      "ts" : {
         "order" : "asc"
      }}]}
```

Dokumentace programu

Pro pravidelné stahování nejnovějších historických dat slouží skripty download.py a process.py, jejichž spouštění je řízeno pomocí bash skriptů download.sh a elastic.sh.
Nutno podotknout, že k přístupu do Vesseltracker Elasticsearch databáze je nutno se nacházet v jejich interní síti, takže mimo síť lze bez tvorby databáze otestovat lze pouze skripty download.py a download.sh.

download.sh

Volá skript download.py se vstupními parametry počáteční čas a koncový čas a přesměruje standardní výstup - příkazy print() do příslušného .log souboru.

elastic.sh

Provádí čtení textového souboru listofdownloaded, který vzniká při stahování nových dat v download.py jako seznam příslušných časů, ze kterých byla data úspěšně stažena v posledním běhu skriptu. Časy z tohoto souboru jsou předávány postupně jako argumenty pro volání Python skriptu process.py. Též dochází k přesměrování standardního výstupu do .log souboru. Nakonec se ještě přejmenuje listofdownloaded, aby nepřekážel při dalším downloadu.
O pravidelnost spouštění obou bash skriptů se stará program Cron - konkrétně toho bylo docíleno manuální úpravou souboru crontab. Oba bash skripty jsou spouštěny 4x denně několik hodin po sobě.

download.py

Stahování dat z webu na základě zadaného počátečního a koncového času - dva systémové vstupní parametry.

Funkce:

downloaddata
Stahuje data z webu na základě zadaného počátečního a koncového času pomocí linoxového příkazu wget a mění výsledné jméno souboru. Konkrétní časy ze kterých byla data úspěšně stažena jsou ukládány do souboru listofdownloaded.
vstup: datetime - čas pro počátek stahování, datetime - koncový čas
výstup: bez výstupu

determineFile
Na základě vstupní hodnoty času vrací nejbližší čas pro extrakci dat a určuje umístění a název souboru, který by měl být stažen.
vstup: datetime - vstupní hodnota času
výstup: string - název složky, string - název souboru, datetime - čas nejbližšího existujícího souboru

filesPresenceCheck
Funkce, která zjišťuje, zda jsou data na webu k dispozici.
vstup: string - název složky, string - název souboru
výstup: bool hodnoty - true pro existenci hledaného souboru, false pro opak

process.py

Zpracovává a extrahuje potřebná data ze stažených souborů pomocí externího nástroje ecCodes, konkrétně funkce grib_get_data, výsledná data nahrává do databáze Elasticsearch, což lze provádět pouze v interní síti Vesseltracker.

Funkce:

determineClosestDateTime

Na základě daného času vrací nejbližší čas pro výběr dat. (3 hodinové intervaly)
vstup: datetime - daný známý čas
výstup: datetime - čas nejbližšího souboru

is_number

Ověřuje, zda je vstupní string číslo.
vstup: string
výstup: bool hodnoty - true pro číslo, false pro cokoliv jiného

longconvert

Převádí zeměpisnou délku ve formátu 0°- 360° na -180°- 180°.
vstup: string - zeměpisná délka
výstup: string - zeměpisná délka

dictextract

Funkce provádějící extrahování dat ze staženého souboru, který je dán vstupním časem a ukládá data do slovníku.
vstup: datetime - čas pro výběr souboru s daty
výstup: slovník - kde klíč je zem.šířka_zem.délka_čas a obsahuje hodnoty: výška vlny, čas, lokace (zem. délka, šířka).

index_to_elastic

Posílá data uložená ve slovníku do databáze Elasticsearch domluveným postupem dle návodu.
vstup: slovník, datetime - čas vložení
výstup: bez výstupu

download_process_past.py

Stahuje a zpracovává starší data, pokud je to zapotřebí. Není spouštěn automaticky.
Skript má tři vstupní systémové parametry - čas začátku a konce ve formátech Rok-měsíc a poté Elasticsearch databázi, do které budou data posílána.
Funkce:

incrementMonth

Funkce přidává měsíc ke vstupnímu datetime
vstup: datetime
výstup: datetime

downloadData
Stahuje data z webu na základě počátečního a koncového času pomocí linux příkazu wget.
vstup: datatime - počáteční čas, datatime - koncový čas
výstup: list - seznam časů stažených dat

isNumber
Kontroluje, zda je vstupní string číslo.
vstup: string
výstup: bool hodnoty - true pro číslo, false pro cokoliv jiného

longConvert
Převádí zeměpisnou délku ve formátu 0°- 360° na -180°- 180°.
vstup: string - zeměpisná délka
výstup: string - zeměpisná délka

dictextract
Funkce provádějící extrahování dat ze staženého souboru, který je dán vstupním časem a ukládá data do slovníku.
vstup: datetime - čas pro výběr souboru s daty
výstup: bez výstupu

index_to_elastic
Posílá data uložená ve slovníku do databáze Elasticsearch domluveným postupem dle návodu.
vstup: slovník, datetime - čas vložení
výstup: bez výstupu

Závěr

Co bude dál? Jaký další postup projekt čeká? Dá se říct, že v současné době jsou práce na back-end části zastaveny, jelikož už není co dělat. Pravidelná služba na stahování historických dat o vlnách byla úspěšně nasazena na serveru, otestována a běží v pořádku. Co se potenciálního využití těchto dat týče, zde již další aktivity jsou plně na zbytku týmu Vesseltracker. Mezi další části postupu patří vytvoření jednoduchého API rozhraní pro přístup k datům z Elasticsearch stejným způsobem jako dosavadní služby tedy s aplikováním přihlašovacích formulářů a autentizaci uživatele apod. Do budoucna se též počítá s implementací volání tohoto přístupového API v nové variantě webové mapové aplikace, na které stále probíhají aktivní práce.

Na tato data lze co se týče charakteru a využití pohlížet jako na bodová data, například úloha zjisti výšku vln poblíž pozice dané lodě před třemi týdny v 8:35. Zde se přímo nabízí propojení se stávající webovou mapovou aplikací. Zároveň by bylo možné pro konkrétní časové úseky vygenerovat izočáry (polygony) pro oblasti s výškou vln nad 5 metrů apod. Tyto vrstvy by pak mohly anebo nemusely být dlaždicovány pro zobrazení jako překryvná vrstva ve výsledné webové mapové oblasti. Dalším možným - spíše teoretickým využitím dat v tomto formátu by bylo charakterizování oblastí dlouhodobě exponovaných působení vysokých vln a nepřízně počasí například formou liniových grafů. Ovšem pro tyto účely by bylo vhodnější využít formát původní formát GRIB a pracovat s ním přímo.

V průběhu tvorby skriptů bylo též zjištěno, že GRIB data lze vcelku snadno publikovat a zobrazovat též pomocí Geoserveru v rámci pluginu pro zobrazování NetCDF rastrů - který umožňuje i formát GRIB. To by umožnilo jejich snadné publikování jako WMS pro potřeby vizualizace ve webové mapové aplikaci s nějakou pevně nastavenou škálou - například do pěti metrů průhledná, poté podle výšky škálované poloprůhledné odstíny šedi.

Rozdělení práce na projektu bylo následující:

Lubomír Bucek - komunikace s Vesseltracker, Python skripty download.py a process.py, bash skripty elastic.sh, download.sh, část dokumentace.

Zuzana Richtrová - Python skript download_process_past.py, část dokumentace, příprava prezentace.