infostat® Inštitút informatiky a štatistiky
DÁVAME INFORMÁCIÁM ZMYSEL

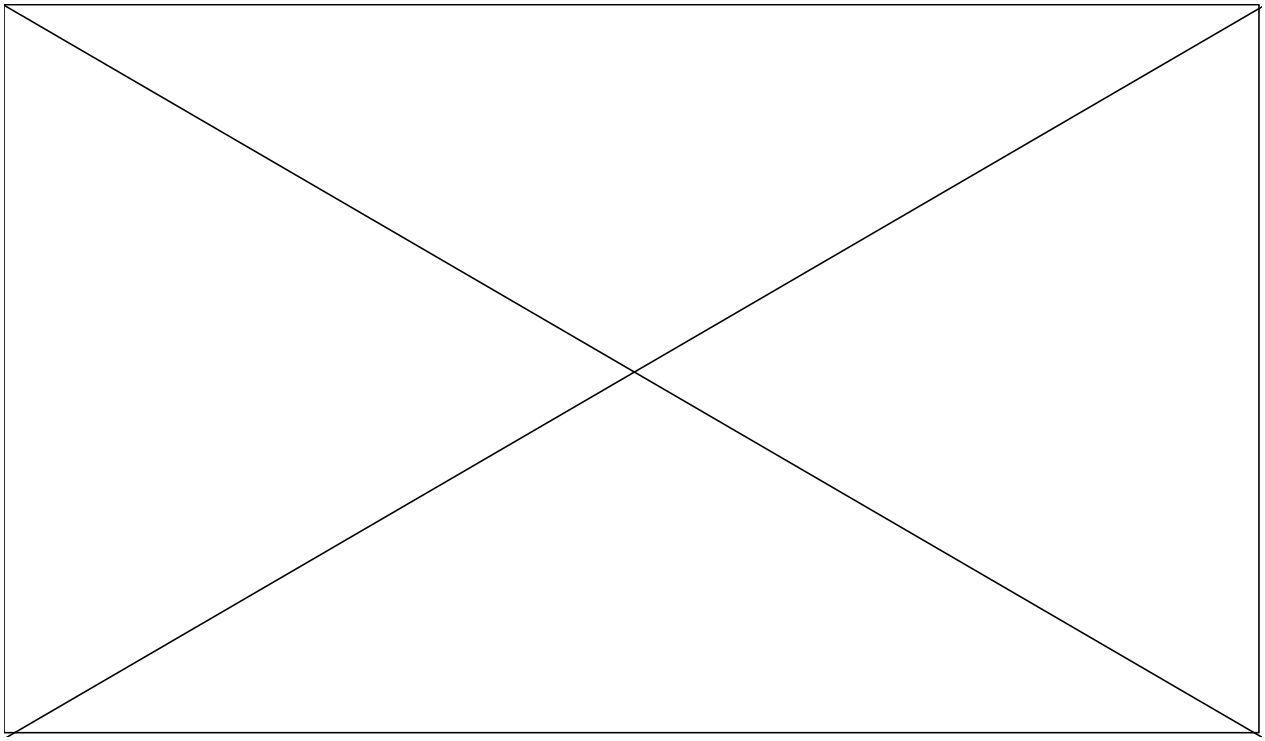
Big Data

Version 5.03_18.12.2014

Robert Suja

Obsah

Zhrnutie



Úvod

Big Data je jedným z vedúcich technologických trendov posledných rokov. Väčšina globálnych spoločnosti v oblasti IT *(a nielen v IT)* už plne integruje Big Data do vlastných IT a DWH infraštruktúr. Oblasťou , ktorá ešte nevyužíva potenciál Big Data je verejný sektor a štátna správa. Vo všeobecnosti štátnej správe a verejnému sektoru stále dominujú tradičné štatistické zdroje, ktoré sú v súčasnosti takmer exkluzívne postavené na prieskumoch a získavaní administratívnych dát zo štátnej a verejnej správy. Isté známky zlepšenia je ,ale možné už postrehnúť aj v tejto oblasti. V rámci EU sa v priebehu minulého roka viedli diskusie na globálnej úrovni o identifikácii možností, ktoré Big Data prinášajú oficiálnej štatistike a zároveň o hlavných strategických a metodických problémoch, ktoré Big Data predstavujú pre oficiálnu štatistiku. Záverom týchto debát bolo Scheveningenske Memorandum :

Eurostat – (CORS - Collaboration in Research and Methodology for Official Statistics)
Scheveningen memorandum plné znenie memoranda

V jednotlivých členských štátoch prebiehali a prebiehajú aj individuálne projekty. Za zmienku stoji predovšetkým Holandsky štatistický úrad, ktorý už ma za sebou niekoľko projektov. List niekoľkých zaujímavých európskych projektov je uvedený v tabuľke.

Štatistický úrad	Názov projektu	Popis
EUROSTAT	Index spotrebiteľských cien tovarov a služieb - zisťovaných na internete.	Vývoj nástroja na monitorovanie spotrebiteľských cien jednotlivých reprezentantov zisťovaných vo vybranej sieti e-shopov a prevádzok služieb poskytovaných na internete. Podobnosť s projektom "The Billion Prices Project" http://bpp.mit.edu/
EUROSTAT	Štatistika cestovného hop ruchu založená na dátach o polohe	Na základe dát poskytnutých mobilnými operátormi sú generované štatistiky o polohe a cestovnom ruchu.
ISTAT	Prieskum o využívaní informačných a komunikačných technológií v podnikoch	Prepojenie dát získaných z internetu pomocou "Web Scraping" a "Text Mining" s výsledkami oficiálneho prieskumu "Prieskum o využívaní informačných a komunikačných technológií v podnikoch"
ISTAT	Sledovanie krátkodobej migrácie na území povodia vodných tokov	Sledovanie pohybu rezidentov a návštevníkov na území povodia vodných tokov *(administratívne definovaných)* pomocou priestorových súradníc mobilných telefónov.
Štatistický úrad Slovinskej Republiky	Populačné štatistiky s využitím priestorových súradníc mobilných telefónov.	Využitie priestorových súradníc mobilných telefónov na zlepšenie oficiálnej štatistiky najmä pokiaľ ide mobilitu obyvateľstva.
Holandský Štatistický Úrad	Na Holandskom Štatistickom úradu Holandska bolo vykonaných niekoľko prípadových štúdií na "Big Data".	Zdroje dát, ktoré boli sledované na vhodnosť použitia pre oficiálnu štatistiku: 1. Záznamy z elektronického riadiaceho systému dopravy, 2. Dáta z mobilných telefónov, 3. Správy v sociálnych médiách.

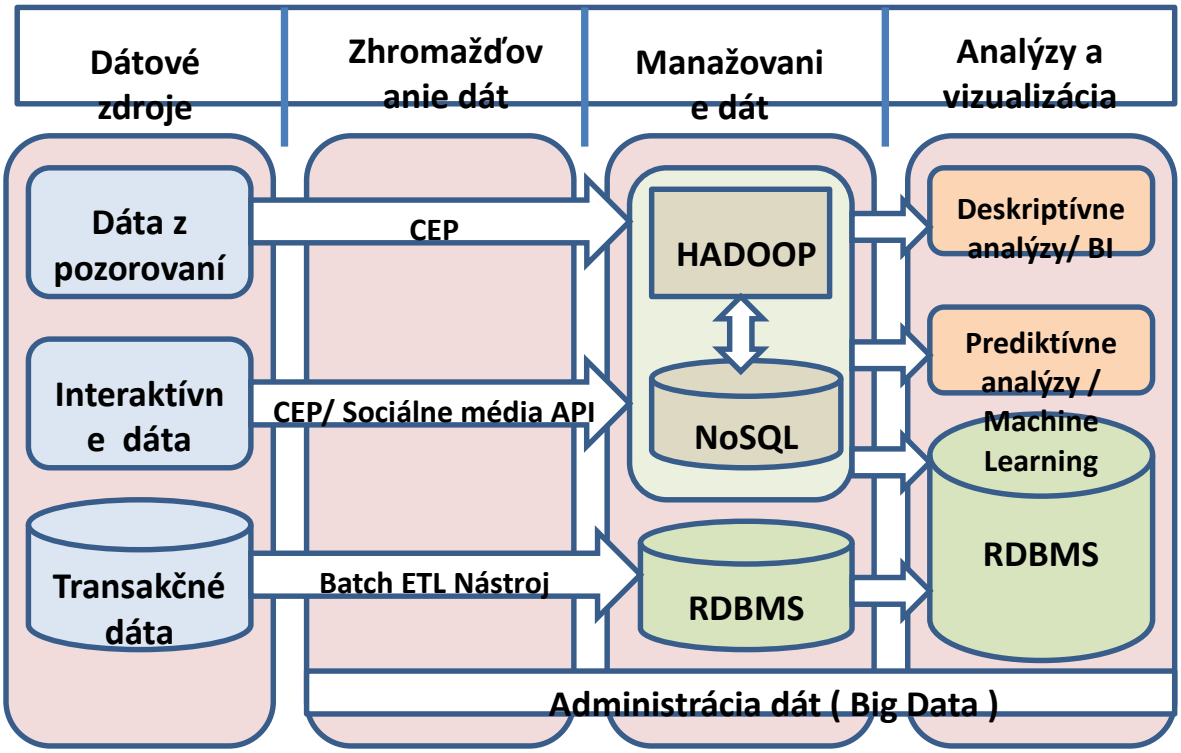
Big Data definícia

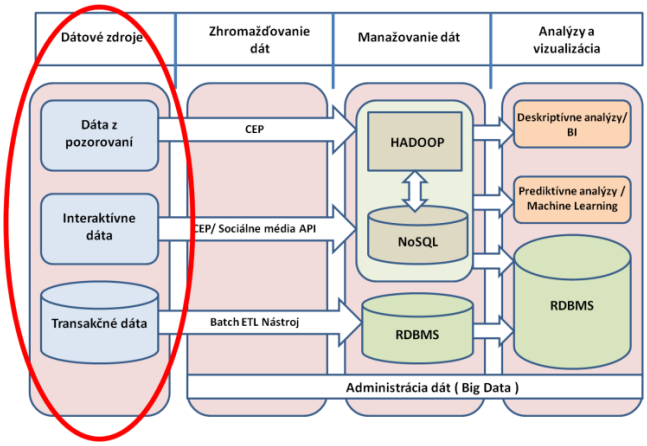
Big Data je koncept, ktorý vznikol na základe obrovského množstva neštruktúrovaných dát ,ktoré denne vznikajú a je ich problém spracovať tradičnými metódami spracovania dát. Existuje niekoľko definícií , ale takmer univerzálne sú vo všetkých spomenuté tri základne charakteristiky Big Data a to sú:

- **Množstvo *(Volume)*** – veľké objemy dát, typicky začínajú na desiatkach terabytov.
- **Rýchlosť *(Velocity)*** – rýchlosť ako sú dáta tvorené poprípade upravované.
- **Pestrosť *(Variety)*** – rôznorodosť dátových formátov

Pokiaľ dáta spĺňajú aspoň dve z uvedených charakteristík a tak môžeme hovoriť o Big Data.

Jednoduchý popis architektúry je znázornený na obrázku.



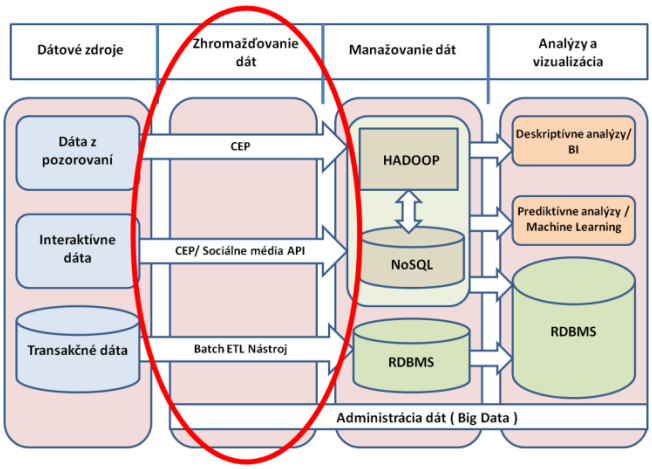


Dátove zdroje v chápaní konceptu Big Data môžeme rozdeliť na:

- *Dáta z pozorovaní* – prístrojovo generované dáta, dáta v reálnom čase napríklad zo senzorov a snímačov a pod.. Vo všeobecnosti je hlavným zdrojom dát z pozorovaní "Internet of Things" alebo „IoT" . „IoT" je nový trend v oblasti kontroly a

komunikácie objektov rôzneho využitia medzi sebou alebo s človekom, a to najmä prostredníctvom technológií bezdátového prenosu dát a internetu.

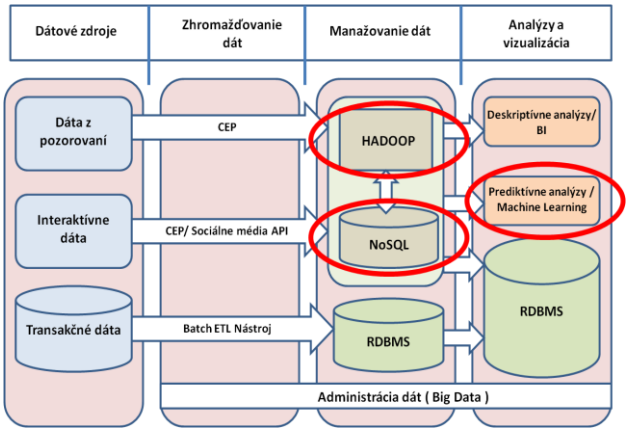
- *Interaktívne dáta* – dáta zo sociálnych sietí napr. LinkedIn, Twitter, Facebook, atd., a sledovania aktivity na internete ako aj Web obsahu – web logs, video, fotky a pod..
- *Transakčné dáta* – dáta popisujúce nejakú udalosť (jej zmenu ako výsledok transakcie). Vždy obsahujú zápis o čase vzniku prípadne zmeny, numerickú hodnotu a referujú na jeden alebo viac objektov. Typickým príkladom sú napríklad: finančné operácie, logistické operácie alebo zápisy pracovnej aktivity.



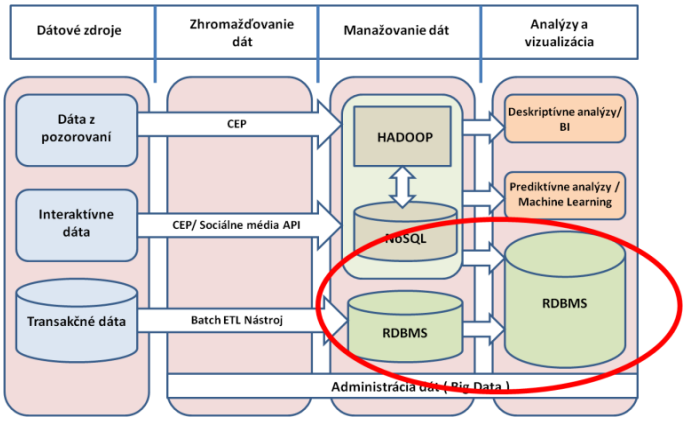
Zhromaždovanie dát v chápaní konceptu Big Data môžeme rozdeliť na:

- *CEP* – Complex event processing (CEP), metóda na sledovanie a sprocesovanie dát z rôznych zdrojov s cieľom identifikovať významné/zaujímavé charaktiristiky a väzby medzi nimi.
- *API* – Application Programming Interface (API) - Definuje vzájomnú komunikáciu jednotlivých softwarových komponentov. Okrem prístupu k DB a počítačovému HW sa používa aj pre prácu v GUI (grafickom rozhraní).

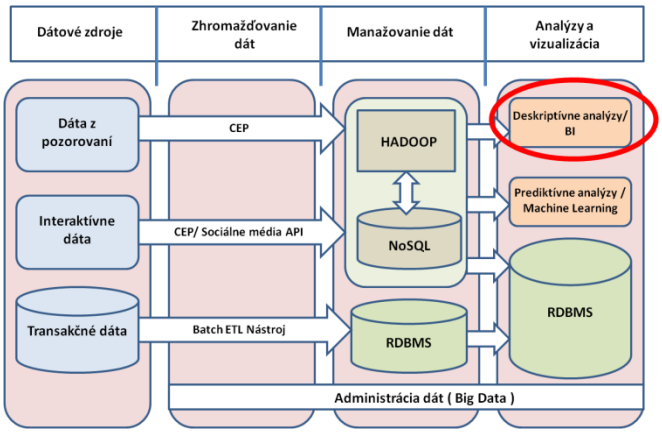
- *ETL* – Extrakcia, transformácia a nahrávanie dát. Extrakcia znamená získanie dát z dátových zdrojov a ich následné uloženie do dočasného úložiska. Transformácia predstavuje proces spracovania získaných primárnych dát do formy odpovedajúcej požiadavkám dátových skladov. Transformácia zahrňuje celú škálu operácií od konverzie, matematických operácií, filtrovania, normalizácie a denormalizácie. Prichádzajúce dáta môžu byť znečistené rôznymi typmi chybných či nekompletných údajov. Súčasťou transformácií preto býva mechanizmus kontroly a čistenia dát. Záverečnou fázou je nahrávanie spracovaných dát do cieľového systému dátového skladu. Batch – Dávkové spracovanie používané u ETL, vykonané sériou programov (dávok) bez priamej účasti užívateľa.



HADOOP, NoSQL a **Machine Learning** predstavujú nosné koncepty práce s Big Data a preto je každej oblasti venovaná samostatná kapitola. Všetky ostatné oblasti uvedené v schéme architektúry boli a sú využívané aj pri tradičnom spracovaní dát *(dát, ktoré nie sú kritériami definované ako Big Data)*.



RDBMS – je databáza založená na relačnom modely. Často sa takto označuje nie len samotná databáza, ale aj jej konkrétne softwarové riešenie. Relačná databáza je založená na tabuľkách, kde riadky obsahujú jednotlivé záznamy a stĺpce obsahujú informácie o reláciách medzi nimi *(primárny kľúč - jednoznačný identifikátor v tabuľke, ktorý reprezentovaný stĺpcom alebo skupinu stĺpcov)*. RDBMS všetky vzťahy reprezentuje vo forme tabuliek, dvojrozmerné tabuľky stačia na modelovanie reálnych vzťahov.



Deskriptívne analýzy/ BI – poskytujú možnosť analyzovania dát, objavovania vzorcov a súvislostí vo veľkých dátových súboroch s cieľom získať znalosti z existujúceho dátového súboru a premeniť ich na štruktúry zrozumiteľné pre človeka. Zároveň umožňujú obohatenie dát, hierarchizáciu a hľadanie závislostí. Je možné organizovať dáta do intuitívnych štruktúr pre podporu preddefinovaných aj jednorazových dotazov, ktoré dokážu identifikovať pravidlá, vzťahy a trendy.

Big Data serializácia

Big Data sú tvorené rôznymi zariadeniami, prechádzajú cez rôzne systémy a uložené sú v rôznych DB. Pracujú s rôznymi programovacími jazykmi a APIs. Z toho dôvodu je dôležitá dátová serializácia. Serializácia znamená prevedenie dátovej štruktúry na jednotnú sériu, ktorú je možné uložiť do DWH, prípadne preniesť po sieti. Okrem samotného prevodu a formátu prevodu je potrebné myslieť aj na prevod referencií k dátam. Pri výbere formátu serializácie je treba počítať aj s prípadným prevodom späť do pôvodného formátu *(deserializácia)*.

JSON - JavaScript Object Notation (*Javascriptový objektový zápis*, **JSON**) je spôsob zápisu nezávislý na počítačovej platforme, určený pre prenos dát, ktoré môžu byť organizované v poliach alebo agregované v objektoch. Vstupom je ľubovoľná dátová štruktúra *(číslo, reťazec, boolean, objekt alebo z nich zložené pole)*, výstupom je vždy reťazec. Zložitosť hierarchie vstupnej premennej nie je nijako obmedzená. Príklad JSON:

```
{"menu": {
  "id": "file",
  "value": "File",
  "popup": {
    "menuitem": [
      {"value": "New", "onclick": "CreateNewDoc()"},
      {"value": "Open", "onclick": "OpenDoc()"},
      {"value": "Close", "onclick": "CloseDoc()"}
    ]
  }
}}
```

BSON - Binary JSON *(BSON)* bol vytvorený MongoDB teamom a stále sa aj používa v rámci MongoDB. BSON reprezentuje akýkoľvek JSON v binárnej forme. Cieľom pri tvorbe BSON nebola zlepšenie zápisu *(rozsah zápisu)* ,ale zrýchlenie procesu konverzie.
Príklad BSON zápisu:

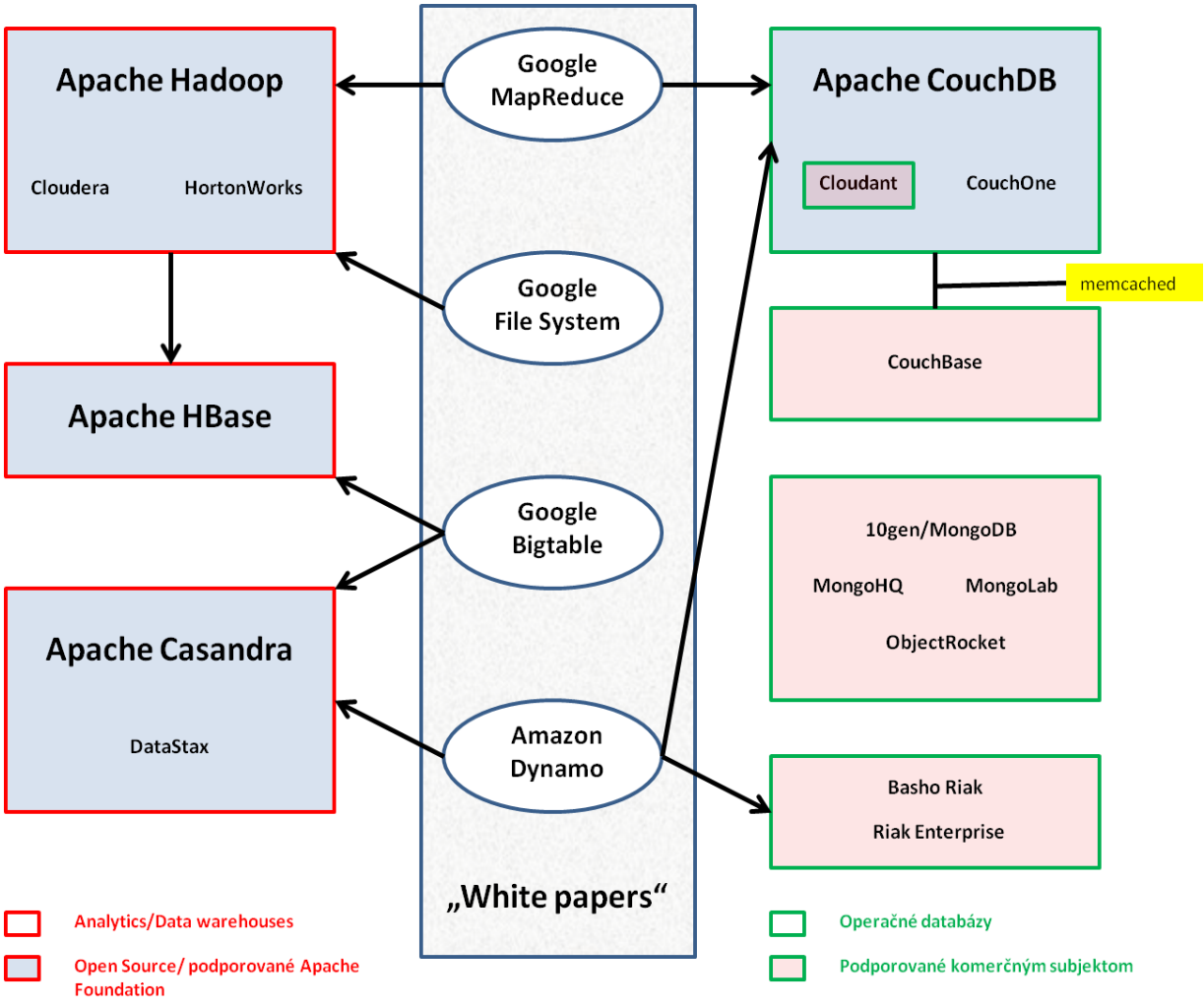
```
{"robert": "suja"} → "\x16\x00\x00\x00\x02robert\x00
                      \x06\x00\x00\x00suja\x00\x00"
```

Apache Thrift – je definícia rozhrania a binárny komunikačný protokol**,** ktorý sa používa na komunikáciu medzi programovacími jazykmi ako C++, Java, Python, PHP, Ruby, Erlang, Perl, Haskell, C#, Cocoa, JavaScript, Node.js, Smalltalk, OCaml, Delphi a iné**.** Bol vytvorený Facebook-om a v súčasnosti patrí pod Apache Foundation. Pomocou Trift sa najskôr preddefinuje dátová štruktúra a štruktúra rozhrania, ktoré sa bude používať na komunikáciu. Trift následne vygeneruje serializačný kód na serializáciu a deserializáciu dát a funkcií.

Avro je koncept na serializáciu dát a RPC *(remote procedure call – je protokol pri ktorom môže program použiť služby programu lokalizovaného na inom počítači, bez potreby hlbšej znalosti siete/prepojenia)*. Ponúka podobnú funkčnosť ako Trift. Používa JSON pre definíciu dátových typov, protokolov a serializáciu dát v binárnom formáte. V kontexte Hadoop môže byť použitý napríklad na prevod z C *(programovací jazyk)* do Pig *(programovací jazyk).*

NoSQL

NoSQL znamená „nie len SQL“ *(Not only SQL)*. NoSQL nedefinuje konkrétnu technológiu alebo riešenie, je všeobecným názvom, ktorý zahŕňa všetky riešenia, ktoré nie sú postavené na báze relačného modelu.
NoSQL vzniklo v internetových gigantoch Google, Amazon a Facebook, ktorí mali problémy spracovať veľké objemy dát tradičnými RDBMS. Na základe ich poznatkov, ktoré boli následne publikované vznikli ďalšie NoSQL DB. Jednoduchý prehľad niektorých NoSQL DB a ich väzieb na „pôvodné“ NoSQL DB je znázornený na obrázku.



Z obrázku je zrejmé veľa súčasných NoSQL DB vzniklo práve na prvých konceptoch NoSQL. Napríklad **Apache CouchDB™** vznikol na podkladoch Google **MapReduce white paper**, and Cloudant na Apache CouchDB a **Amazon's Dynamo white paper** . Iné napríklad MongoDB vznikli nezávisle od uvedených konceptov.

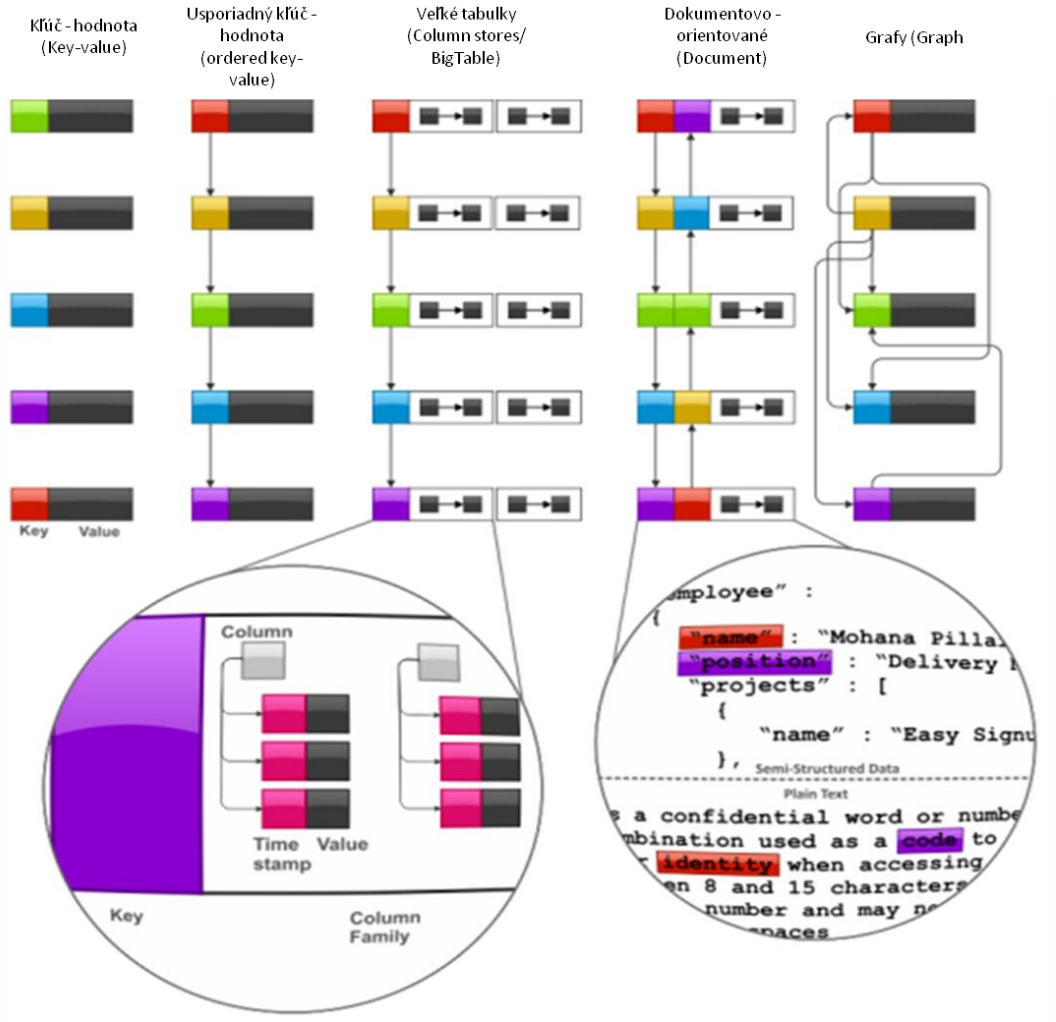
Spracované dáta v NoSQL môžu byť štruktúrované, neštruktúrované, alebo semi-štrukturované. Využíva schopnosť ukladať a načítať veľké množstvo dát bez závislosti na vzťahoch. Pracuje sa s

distribuovanou architektúrou a s dátami, ktoré sú redundantným spôsobom uložené na niekoľkých serveroch. Týmto spôsobom je možné ľahko škálovať systém, pridávaním ďalších serverov, tým vzniká tolerancia k prípadným výpadkom. Fakt, že majú jednoduchú štruktúru a model umožňuje jednoduché zálohovanie a paralelné výpočty.

Z pohľadu spôsobu ukladania dát *(dátového modelu)* sa dajú rozdeliť na:

- Úložisko kľúč - hodnota *(Key-value stores)*,
- Implementácia veľkých tabuliek *(Column stores/ BigTable),*
- Dokumentovo - orientované *(Document databases),*
- Grafové databázy *(Graph databases).*

Schematické znázornenie rozdelenia podľa dátových modelov:

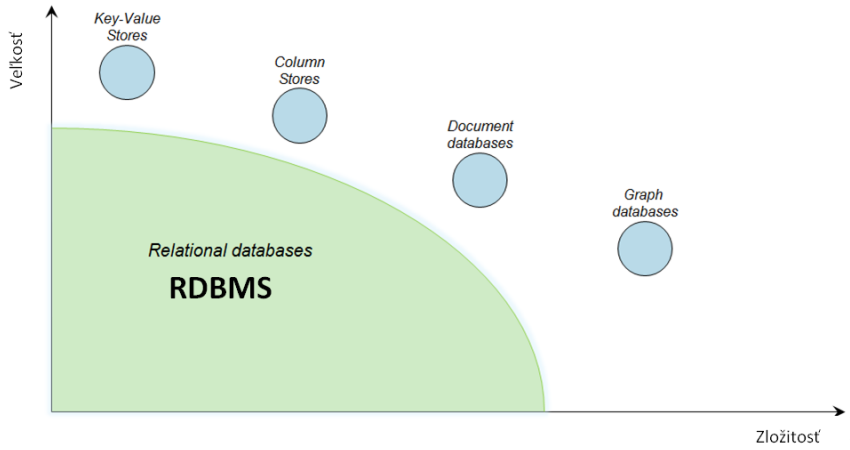


Rozdiel medzi relačným modelom a NoSQL dátovým modelom:

- Relačný dátový model sa zaujíma o štruktúru dostupných dát -> Aké odpovede mám?
- NoSQL model sa zaujíma o formu prístupu k dátam -> Aké otázky mám?

- NoSQL model si často vyžaduje hlbšiu znalosť dátových štruktúr a algoritmov ako relačný model
- Relačné DB nie sú vhodné pre hierarchické alebo grafické dátové modely

Využiteľnosť jednotlivých DB podľa objemu spracovaných dát a ich náročnosti na spracovanie.



Základné princípy NoSQL dátového modelingu

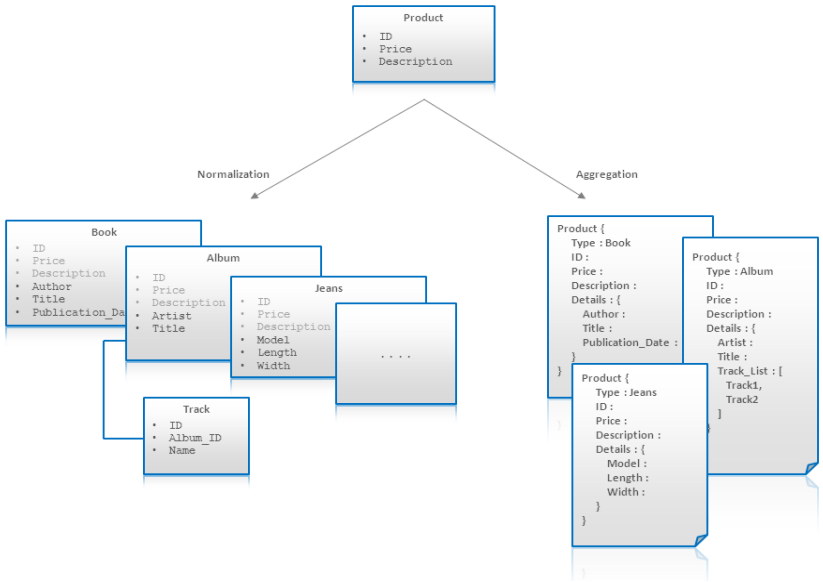
Denormalizácia – je možné definovať ako kopírovanie dát do viacerých dokumentov alebo tabuliek s cieľom zjednodušiť vyhľadávací proces.
Používa sa v: Úložiskách kľúč - hodnota *(Key-value stores)*, Dokumentovo - orientovaných DB *(Document databases)*, Veľkých tabuľkách *(Column stores/ BigTable implementation)*

Agregácia – zhlukovanie dát. Tento model používajú všetky NoSQL modely.

Úložisko kľúč - hodnota (Key-value stores) a Grafové databázy (Graph databases) - typicky nekladú žiadne obmedzenia na dátové hodnoty. Hodnoty môžu byť uložené v ľubovoľnom formáte pod jedným kľúčom napr. meno, email, správy pod kľúčom UserID vo formáte UserID_meno, UserID_email, UserID_správy.

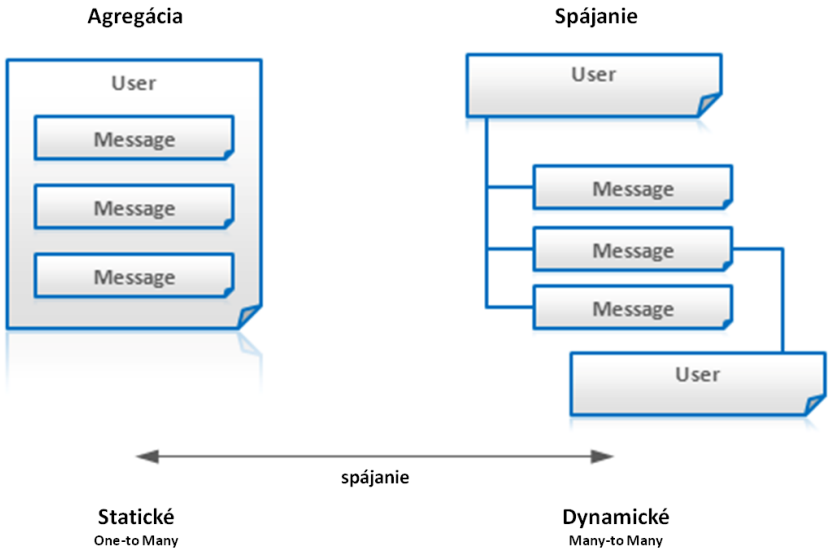
Big Table – podporuje zlúčenie viacerých stĺpcov pod rodinu stĺpcov

Dokumentovo - orientované (Document databases) – umožňujú ukladanie prichádzajúcich dokumentov podľa užívateľa definovanej schémy.



Agregácia umožňuje vytváranie komplexných interných dátových štruktúr. Minimalizáciu dátových väzieb a vzťahov. Porovnanie agregácie a normalizácie je znázornené na obrázku.

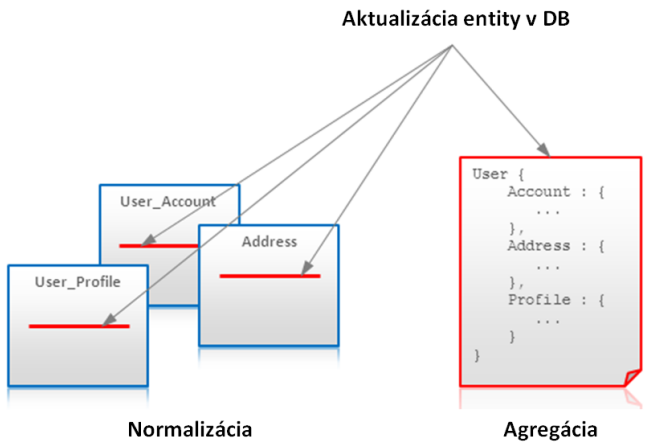
Aplikácia bočného spájania – spájanie dvoch alebo viacerých riadkov do tabuľky alebo jedného riadku je v rámci NoSQL zriedkavé. V NoSQL je možné spájanie nahradiť agregáciou alebo denormalizáciou, ale v niektorých prípadoch to nie je možne. Vtedy je potrebné použiť aplikáciu bočného spájania najmä pri spájaní veľkého počtu vzťahov alebo keď je entita objektu pomerne často modifikovaná. Príklad porovnania agregácie a aplikácie bočného spájania je znázornený na obrázku, kde user je entita pri ktorej dochádza pomerne často v internej modifikácii *(kvôli prichádzajúcim novým správam – message).*



Používa sa v: Úložiskách kľúč - hodnota *(Key-value stores)*, Dokumentovo - orientovaných DB *(Document databases)*, Veľkých tabuľkách *(Column stores/ BigTable implementation)*, Grafových DB *(Graph databases).*

Atomická agregácia – väčšina NoSQL riešení ma limitovanú transakčnú podporu. V niektorých prípadoch je možné dosiahnuť transakčné správanie pomocou súbežného riadenia MVCC *(každý*

užívateľ vidí iba snímky DB spravené v určitom čase, ktoré sa priebežne aktualizujú) alebo použitím agregácie , ktorá môže garantovať niektoré ACID vlastnosti *(ACID - **A**tomicity, **C**onsistency, **I**solation, **D**urability, ktoré garantujú spoľahlivé sprocesovanie transakcií)*.



Rozdiel v aktualizácii medzi relačnou DB *(normalizácia)* a NoSQL *(agregácia)* je znázornený na obrázku. Relačná DB zvyčajne vyžaduje aktualizáciu v niekoľkých tabuľkách, zatiaľ čo NoSQL pomocou agregácie dovoľuje uložiť aktualizáciu ako jeden dokument, riadok alebo kľúč – hodnota pár *(atomická agregácia)*.

Používa sa v: Úložiskách kľúč - hodnota *(Key-value stores)*, Dokumentovo - orientovaných DB *(Document databases)*, Veľkých tabuľkách *(Column stores/ BigTable implementation)*

Kľúče triedenia – jedna z najväčších výhod úložiska kľúč – hodnota je, že jednotlivé dáta môžu byť rozdelené na niekoľkých serveroch iba pomocou zatrieďovacieho kľúča.
Napríklad pri triedení emailov umožňujú niektoré NoSQL DB generovať sekvenčné *ID* čo poskytuje možnosť uložiť jednotlivé správy ako kompozitné kľúče *správaID, užívateľID,* táto vlastnosť nám umožňuje plynulo prechádzať jednotlivé správy/ užívateľov pomocou ich prideleného ID. Správy môžu byť ďalej zgrupované podľa dátumov alebo iných kritérií.

Používa sa v: Úložiskách kľúč - hodnota *(Key-value stores)*

Popis niektorých NoSQL DB

MongoDB

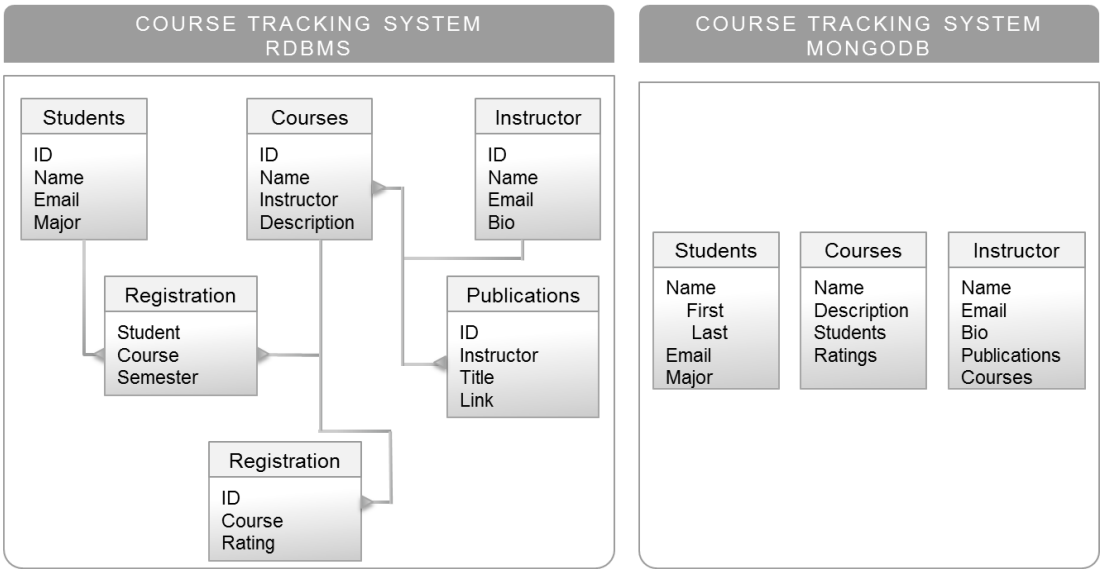
MongoDB *(názov pochádza zo slova "hu**mongo**us" -> „enormný“)* je voľne dostupná dokumentovo - orientovaná databáza. Podporovaná Apache Foundation.
Jednotlivé záznamy v DB sú vnímane ako dokument s vlastným ID a typovým označením *(operácie nad ním sú atomické)*. Dokumenty sú ukladané do zbierok. Jedna zbierka môže obsahovať ľubovoľný počet dokumentov. MongoDB ukladá štruktúrované dáta JSON *(JavaScriptObjectNotation)* ako dokumenty s dynamickými schémami. Dáta sú uložené a von prezentované ako formát BSON, čo je špecifikácia pre prevod JSON do binárnej podoby. Dátový model je veľmi flexibilný, každé políčko môže obsahovať viacej hodnôt. V políčku môže byť uložená aj ďalšia dátová štruktúra. Políčka a ich hodnoty nie sú povinné.
MongoDB podporuje:

- Ad hoc dotazy – má vlastný dotazovací jazyk podobný SQL. Dotazy môžu vrátiť špecifické polia dokumentov a tiež užívateľom definované JavaScript funkcie.
- MapReduce – ako návrhový vzor je implementovaný pre hromadné operácie nad dátami a využíva JavaScript.

- Indexovanie - akékoľvek pole v dokumente môže byť indexované. K dispozícii je aj sekundárne indexovanie.
- Replikácia – je podporovaná master-slave replikácia. Master číta a zapisuje. Slave kopíruje dáta z Master,dáta môžu byť použité iba na čítanie alebo zálohovanie. Slave má možnosť vybrať iného Master ak je predchádzajúci nedostupný.
- Load balancing MongoDB - podporuje horizontále škálovanie. Vyberie sa fragment - kľúč *(shard key)*, ktorý určí ako budú dáta distribuované. Dáta sú rozdelené a distribuované na viacej fragmentov.

Pre vyváženie zaťaženia MongoDB môže bežať na viacerých serveroch a s duplikáciou dát v prípade, ak by niektorý server zlyhal.

Porovnanie tradičnej RDBMS a MongoDB *(zdroj)*



Návod na inštaláciu MongoDB *(iba v anglickom jazyku/ctrl+klik)*

CouchDB

CouchDB je dokumentovo - orientovaná DB s rozhraním JavaScript, podporovaná Apache Foundation a v mnohých ohľadoch podobná MongoDB. Odlišuje sa v podpore dotazovania, škálovania a replikácie. Hlavné rozdiely sú uvedené v tabuľke.

	CouchDB	MongoDB
Dátový model	Dokumentovo-orientovaná (JSON)	Dokumentovo-orientovaná (BSON)
Interface	HTTP/REST	TCP/IP
Skladisko	DB obsahuje dokumenty	DB obsahuje dokumenty a zbierky dokumentov
Query (metódy)	Map/Reduce (javascipt), vytváranie pohľadov + intervalových pohľadov	Map/Reduce (javascipt), vytváranie zbierok + objektovo orientované dotazy
Replikácia	Master-Master s funkciou riesenia konfliktov	Master-Slave
Naprogramované v:	Erlang	C++

Je to databáza pre webové aplikácie a zároveň je ideálnym riešením pre použitie v mobilných zariadeniach najmä vďaka kvôli replikačným a synchronizačným vlastnostiam. Operácie sú vykonávané

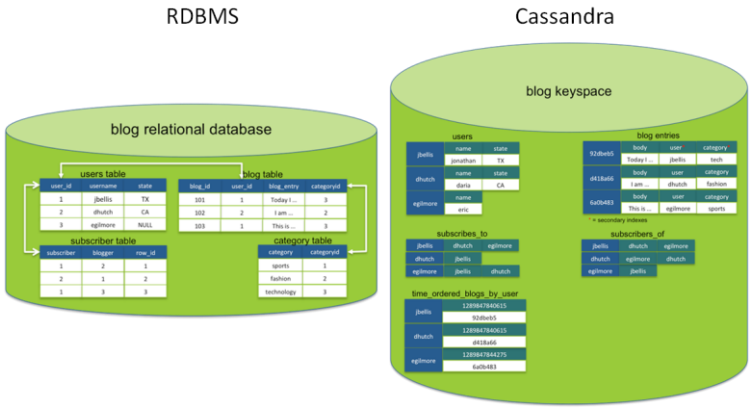
v kontexte dokumentu a dosiahnutie výslednej zhody dát medzi uzlami je zabezpečené pomocou postupnej replikácie. Zmeny dokumentu sú pravidelne kopírované medzi servermi. V prípade neúspešnej replikácie dokáže databáza naviazať na mieste, kde predchádzajúci pokus skončil a pri následnej synchronizácii sa prenášajú iba tie dáta, ktoré sa zmenili. CouchDB je inštalovaná v mobilných telefónoch a ďalších mobilných zariadeniach tak, že sa synchronizuje s centrálnou CouchDB kedykoľvek je online.
CouchDB podporuje:

- B-tree úložiská - B-tree je utriedená dátová štruktúra, ktorá umožňuje hľadanie, vloženie a mazanie dát. CouchDB používa B-tree pre všetky dáta, dokumenty a pohľady.
- MapReduce - algoritmus je používaný k získaniu pohľadu.
- Indexovanie - dokumenty sú indexované v B-tree podľa ich názvu a sekvencie ID. Každá aktualizácia na inštanciu databázy generuje nové poradové číslo. Sekvencie ID slúžia neskôr na hľadanie zmien v databáze.

Návod na inštaláciu CouchDB *(iba v anglickom jazyku/ctrl+klik)*

Cassandra

Cassandra je založená na Google BigTable a pôvodne sa jednalo o interný projekt Facebook-u, z ktorého sa neskôr stal open source projekt podporovaný Apache Foundation. V DB Cassandra sú dáta uložené ako viacrozmerné mapy.
Každý dátový záznam obsahuje kľúč priradený k určitým hodnotám. Hodnoty sú zoskupené do rôznych „rodín stĺpcov“ *(column-family)*. Každá rodina stĺpcov obsahuje mapu dát. Dáta sú ukladané do riadkov, ktoré majú viacero stĺpcov a sú združené s kľúčom riadka. Každý riadok má voľnú schému, takže môže obsahovať rôzne stĺpce. Každá bunka *(obsah stĺpca pre daný kľúč riadku)* okrem samotnej hodnoty obsahuje aj časovú známku *(timestamp)*, ktorá slúži na odhaľovanie prípadných konfliktoch pri zápise. Pre jednoduchšie vysvetlenie porovnanie CassandraDB s RDBMS *(vid. obrázok).*



(zdroj)

Riešenie Cassandra je určené pre spracovanie veľkých dátových objemov pracovne vyťažených cez viac uzlov, bez jediného bodu zlyhania. Prípadný problém zlyhania rieši Cassandra pomocou peer-to-peer distribuovaného systému, kde sú všetky uzly rovnaké a dáta sú medzi ne rozdelené.

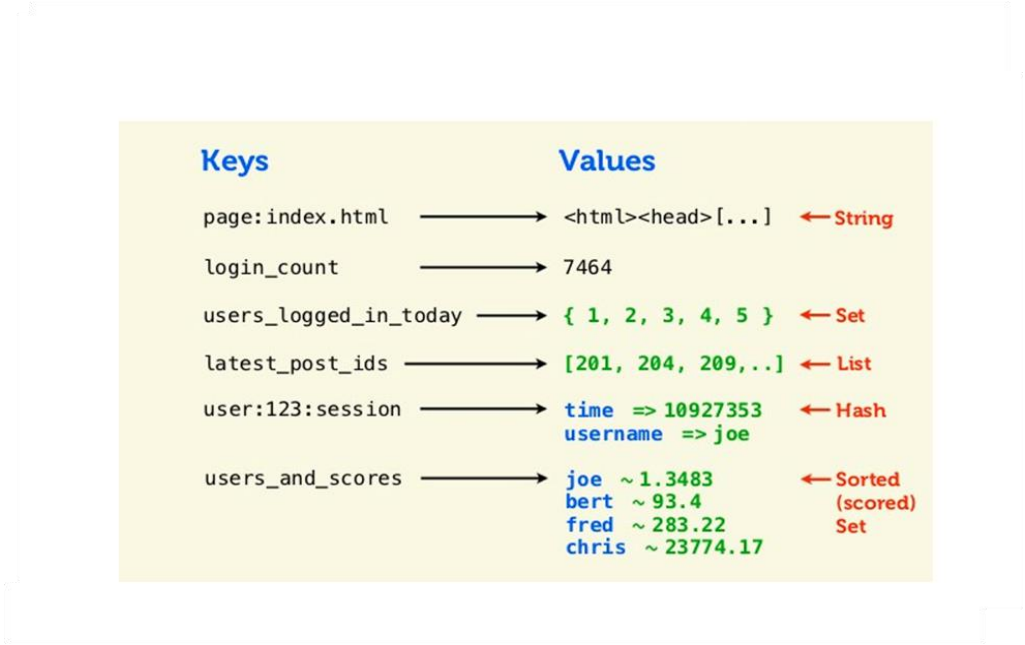
Návod na inštaláciu Cassandra *(iba v anglickom jazyku/ctrl+klik)*

Redis

Redis sa vyznačuje dvoma charakteristickými znakmi: 1.celá DB je uložená v RAM a 2.uložené dáta môžu mať zložitú štruktúru. Tieto dve charakteristiky ponúkajú rýchly a stabilný výkon pri práci s DB, ale rýchlosť výrazne ustupuje v prípade, že expanzia dát na disku začína dosahovať svoje limity. Z toho dôvodu je Redis vhodný na spracovanie menších a do budúcna odhadnuteľných objemov dát. Zároveň ponúka veľmi pôsobivý výkon pri spracovaní zložitých dátových štruktúr čo už do rýchlosti, alebo pestrosti spracovania dát.

Redis je úložisko kľúč - hodnota (Key-value stores) a základné dátové typy sú *(grafické znázornenie na obrázku)*:

- Strings
- Lists
- Sets
- Sorted/Scored sets
- Hashes



(zdroj)

Návod na inštaláciu Redis *(iba v anglickom jazyku/ctrl+klik)*

BigTable

Big Table je pruktom Google a nie je voľne dostupný, jeho časť je prístupná vývojárom na Google App Engine. Má komplexnejšiu štruktúru a rozhranie ako väčšina NoSQL DB, s hierarchickým a multidimenzionálnym vstupom. Prvá úroveň tak ako pri tradičných RDBMS je tabuľka obsahujúca dáta. Každá tabuľka je rozdelená medzi riadky, kde každý riadok je venovaný špecifickému dátovému reťazcu *(unique key string)*. Hodnoty v rámci riadka sú zoradené do buniek, kde každá bunka je identifikovaná identifikátorom rodiny stĺpcov *(column family identifier)* , názvom stĺpca *(column name)* a časovou známkou *(timestamp).* Jednotlivé riadky sú uložené v dávkach vzostupnom poradí v rámci súborov nazývaných „shards“ , to zabezpečuje efektívny a prehľadný prístup k dátam.

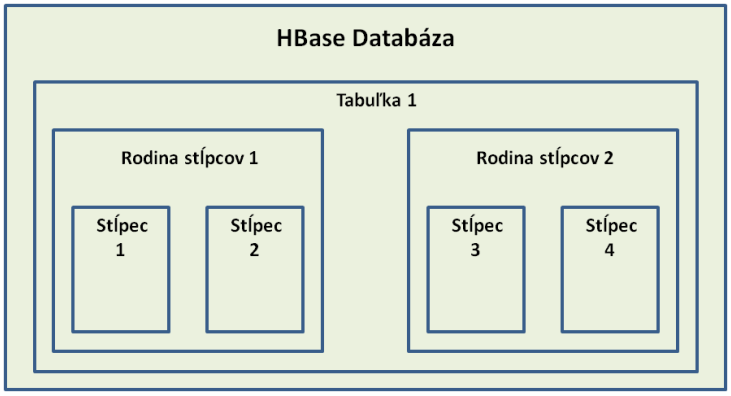
Príklad: doména s menom *com.google.maps/index.html* bude tabuľke uložená v blízkosti *com.google.www/index.html*.

Rodina stĺpcov je niečo ako typ a lebo trieda v programovacích jazykoch , každá reprezentuje skupinu dát, ktoré majú spoločné vlastnosti a charakteristiku napr. v jednej rodine stĺpcov môže byť uložený iba obsah HTML webových stránok. V tabuľke je niekoľko rodín stĺpcov *(ich počet by nemal byť príliš vysoký)* a nemali by sa modifikovať veľmi často. Názvy stĺpcov sú dynamicky definované *(často majú v názve dáta, ktoré daný stĺpec obsahuje)* čo je opakom RDBMS, ktoré majú názvy definované v predstihu. Ak rodina stĺpcov reprezentuje linky k webstránke, názov stĺpca môže byt URL stránky a jednotlivých bunkách je text linky.

Google Big Table je navrhnutý pracovať s veľkými objemami dát, ktoré rozkladá do počítačových klastrov *(commodity hardware).* Garantuje možnosť výberu riadkových transakcií *(možnosť práce s individuálnym riadkom)*, ale neponúka možnosť modifikácie väčšieho počtu riadkov naraz *(nemá vlastnosť „atomicity – buď sa udejú všetky operácie alebo žiadna“).* Pracuje s GFS – Google File System , ktorý priebežne uchováva kópie súborov, pre prípad potreby.

HBase

HBase bola navrhnutá ako „open source“ klon Google BigTable, ktorá sa spolieha na HDFS *(Hadoop Distribution File System, ktorý je v podstate klonom Google File System)*. Podporuje rovnakú dátovú štruktúru tabuliek, stĺpce rodiny (column families), názvy stĺpcov a časové zápisy *(timestamps)*. Vnútorná štruktúra HBase je znázornená na obrázku.



HBase je veľmi dobre integrovaná v rámci projektu Hadoop a ľahko načítava a zapisuje dáta z MapReduce.

Nevýhodou je, že v niektorých prípadoch latencia pri individuálnych pokynoch načítavania a zápisu môže byt pomerne vysoká *(čo spomaľuje spracovanie dát).*

Návod na inštaláciu HBase *(iba v anglickom jazyku/ctrl+klik)*

Hadoop

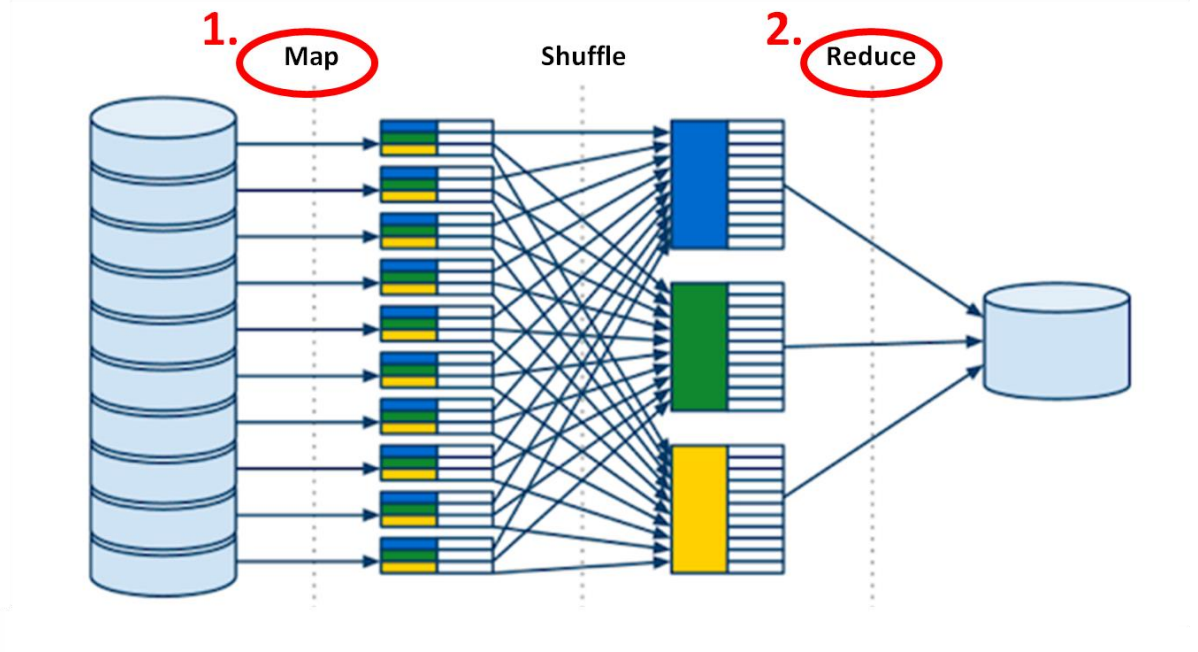
Hadoop je softwarový open source framework podporovaný Apache Foundation, ktorý umožňuje distribuované spracovanie veľkých množstiev dát použitím jednoduchých programovacích modelov. Má schopnosť škálovať z jediného až na tisíce počítačov/serverov, z ktorých každý prispieva výpočtovým výkonom a úložným priestorom. Zároveň zabezpečuje vysokú dostupnosť *(high-avaiability)* dát, pričom dostupnosť nie je zabezpečená na hardvérovej, ale na softvérovej úrovni Hadoop.
V súčasnosti sa Apache Hadoop Framework skladá z nasledujúcich modulov:

- *Hadoop Common* – obsahuje knižnice a služby potrebné k prevádzke iných modulov v rámci Hadoop *(Mirror of Apache Hadoop common)*
- *Hadoop Distributed File System (HDFS)* – distribuovaný systém súborov, ktorý zabezpečuje rozloženie súborov na úložiskách (klastre serverov/počítačov)
- *Hadoop YARN* – riadiaca platforma na koordináciu jednotlivých zdrojov, funkcií (job-ov) a rozvrhu užívateľských aplikácií
- *Hadoop MapReduce* – programovací model pre spracovanie veľkých objemov dát

Apache Hadoop MapReduce and HDFS vychádzajú z Google MapReduce a Google File System (GFS)

MapReduce

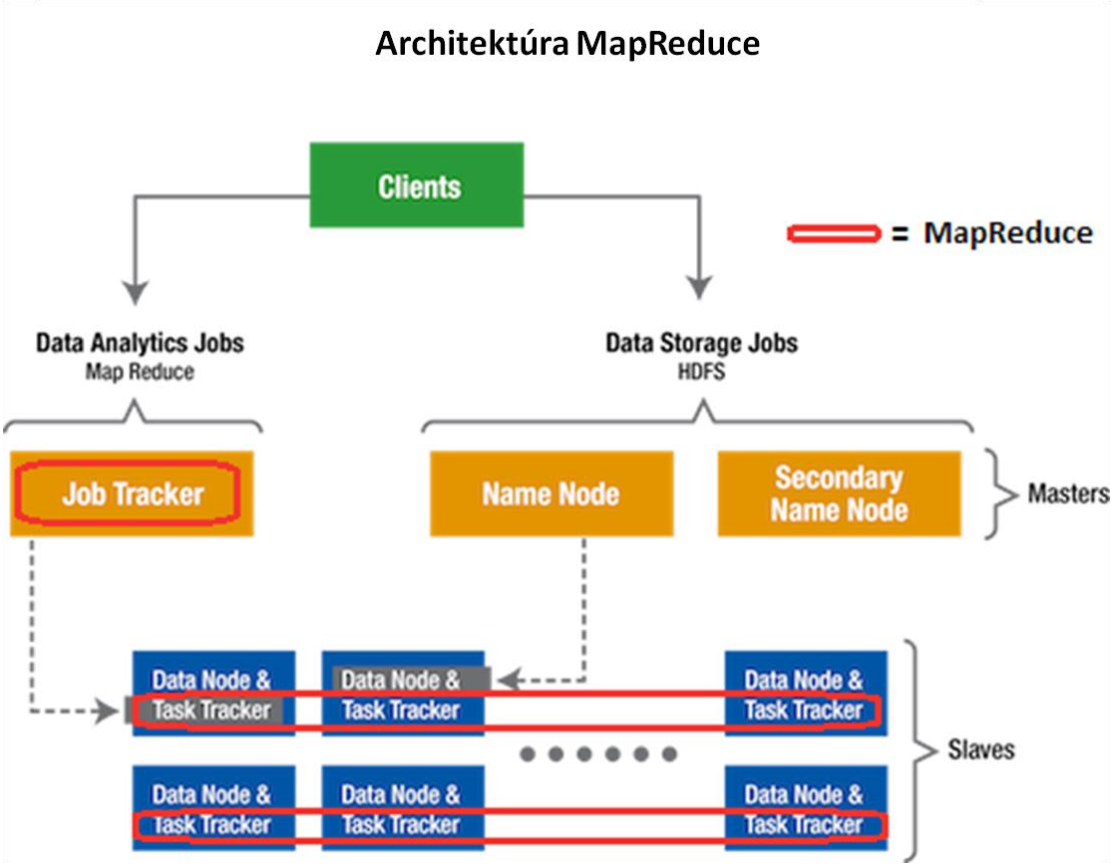
MapReduce je „srdcom“ Hadoop. MapReduce sa skladá z dvoch konzekventných krokov *(viď. obrázok)*:



1. Map krok: Zabezpečuje vstup ,rozdelí ho na menšie čiastkové problémy, ktoré distribuuje do pracovných uzlov. Pracovný uzol môže danú operáciu vykonať znova čo vedie k multi-level stromovej štruktúre. Pracovný uzol spracováva menšiu úlohu, a odovzdá odpoveď späť do svojho hlavného uzla . *Map vykonáva filtrovanie a triedenie.*

2. Reduce krok: Zhromažďuje odpovede na všetky čiastkové úlohy, ktoré kombinuje a vytvorí výstup - odpoveď na problém, ktorý rieši. *Reduce vykonáva sumarizujúcu operáciu.*

Z pohľadu architektúry MapReduce sú procesy rozdelené medzi dve aplikácie JobTracker a TaskTracker. JobTracker beží iba jeden uzol *(node)* klastra, zatiaľ čo TaskTracker beží každého „slave“ uzol v klastri. Každý „MapReduce job“ je rozdelený do niekoľkých čiastkových úloh z ktorých každá je pridelená určitému TaskTracker v závislosti od toho na akom uzli sú dáta uložené. JobTracker je zodpovedný za riadenie zdrojov v klastri a rozvrh jednotlivých vykonávaných činností, zároveň monitoruje progres každého TaskTracker v plnení jeho individuálnych úloh. Grafické znázornenie architektúry MapReduce je na obrázku.



Veľmi jednoduchý príklad na vysvetlenie princípu MapReduce:

Predpokladajme, že máme 5 súborov a každý obsahuje dáta o dvoch stĺpcoch *(key and value)*, ktoré reprezentujú názov mesta a priemernú teplotu v rôzne dni merania.

Príklad dát uložených v jednotlivých súboroch:

Bratislava, 20
Košice, 25
Prešov, 22
Senec, 32
Bratislava, 4
Senec, 33
Prešov, 18

(Samozrejme v reálnych podmienkach budú v aplikácií milióny popr. miliardy riadkov a pravdepodobne nebudú ani naformátované)

Z uvedených dát chceme nájsť maximálnu teplotu pre každé mesto použitím MapReduce.

V prvom kroku priradíme ku každému súboru jednu Map úlohu – *Mapper* a každý Mapper prechádza dáta v danom súbore a vracia iba maximálnu teplotu pre každé mesto. Príklad výstupu:

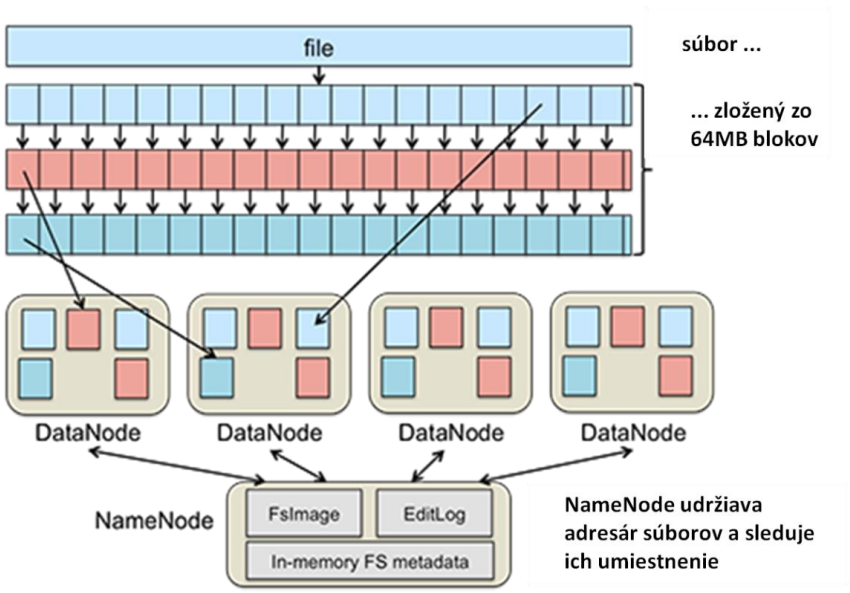
(Bratislava, 20) (Košice, 25) (Prešov, 22) (Senec, 33)

Následne výstupy zo všetkých piatich Mappers prechádzajú do Reduce fázy, kde sa spájajú a následne získavame porovnaním jednotlivých výsledkov konečné hodnoty pre jednotlivé mestá.

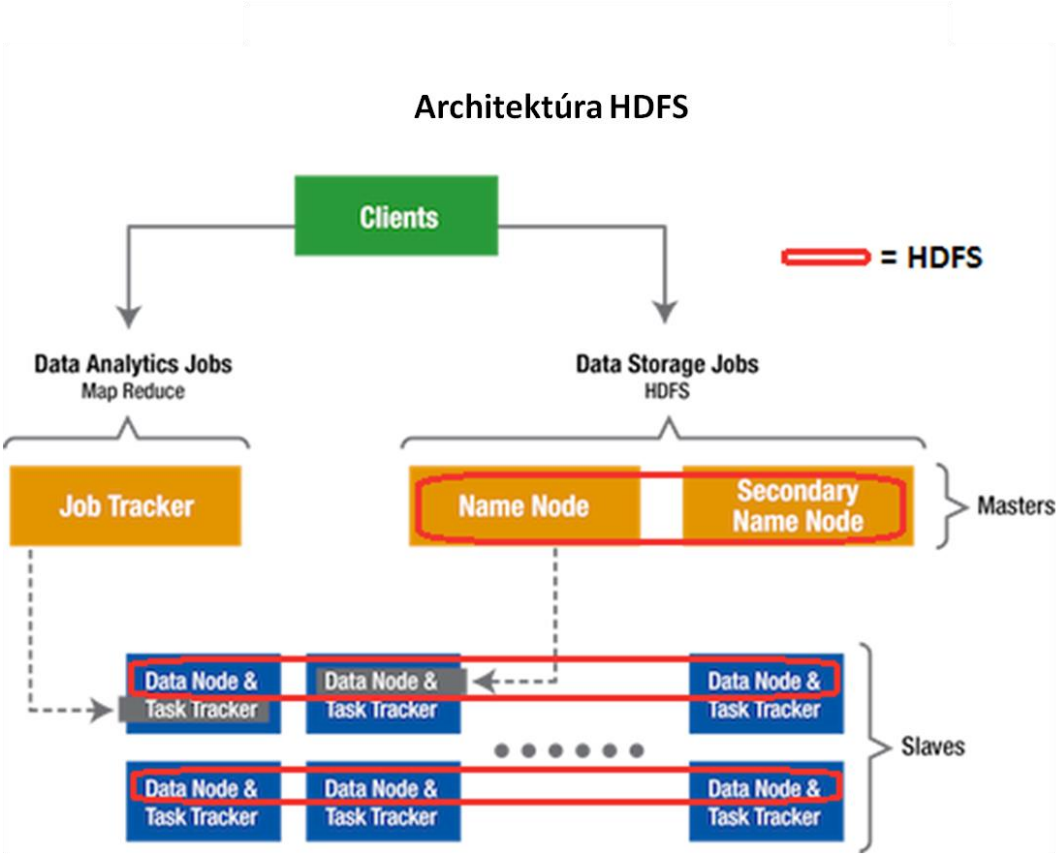
(Bratislava, 32) (Košice, 27) (Prešov, 33) (Senec, 38)

Hadoop Distributed File System (HDFS)

HDFS je to distribuovaný súborový systém určený pre Hadoop, ale použiteľný aj samostatne. Je dimenzovaný na vysokú priepustnosť za zníženej latencie. Typické súbory sú v gigabajtových veľkostiach. Súbory sú rozdelené do blokov o veľkostiach typicky 64 a 128 MB. Bloky sú replikované na viacerých uzloch - DataNode *(zvyčajne 3 násobná replikácia)* viď. obrázok. Primárne bol systém určený iba na zápis, ale neskôr (HDFS-265) sa umožnilo aj pridávanie na koniec súboru *(append).* Tento systém má uzly na rôznych počítačoch prepojených sieťou. Je teda odolný aj voči výpadkom PC a internetu *(ak máme pripojený dostatočný počet PC).*



Z pohľadu architektúry HDFS je hlavným komponentom NameNode, ktorý je zodpovedný manažment metadát a rozloženie súborov v rámci úložiska. Druhým komponentom je DataNode, ktorý slúži k sprocesovaniu dát. Po načítaní dát do HFDS sú dáta replikované a rozdelené do blokov, ktoré sú následne distribuované do DataNodes. NameNode beží na jednom uzli *(node)* klastra a DataNode beží na „slave“ uzli. Grafické znázornenie architektúry je uvedené na obrázku.

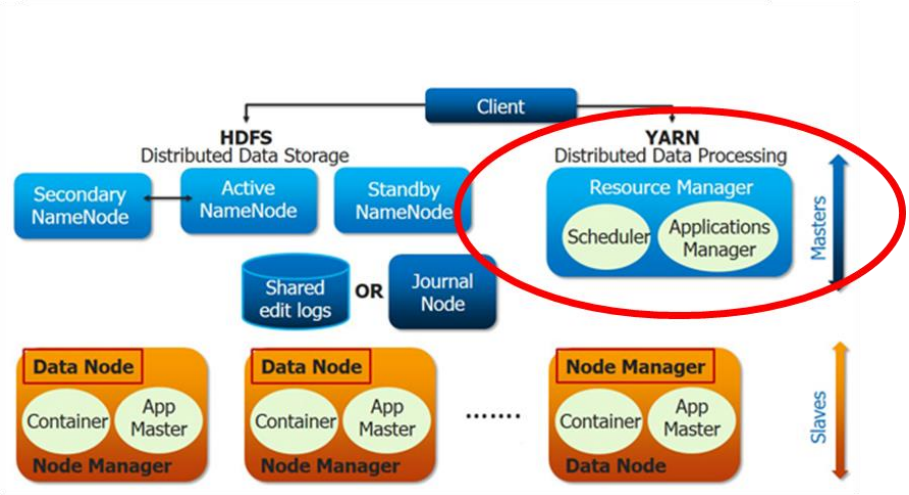


Slabším miestom HDFS, za ktorý bol Hadoop pomerne dosť kritizovaný je „bod zlyhania“ *(SPOF - single point of failure SPOF)*. SPOF leží v NameNode a v prípade, že NameNode alebo server mimo prevádzky tak HFDS je nefunkčné pre celý klaster *(NameNode má aj Secondary NameNode, ktorý robí pravidelné snímky NameNode, ale neslúži ako samotný backup hlavného NameNode)*.

Niekoľko dodávateľov HDFS riešení rieší tento problém pričom zatiaľ asi najobsiahlejšie riešenie tohto problému prišlo od MapR a Cloudera, ktorí patria k najväčších distribútorom Hadoop.

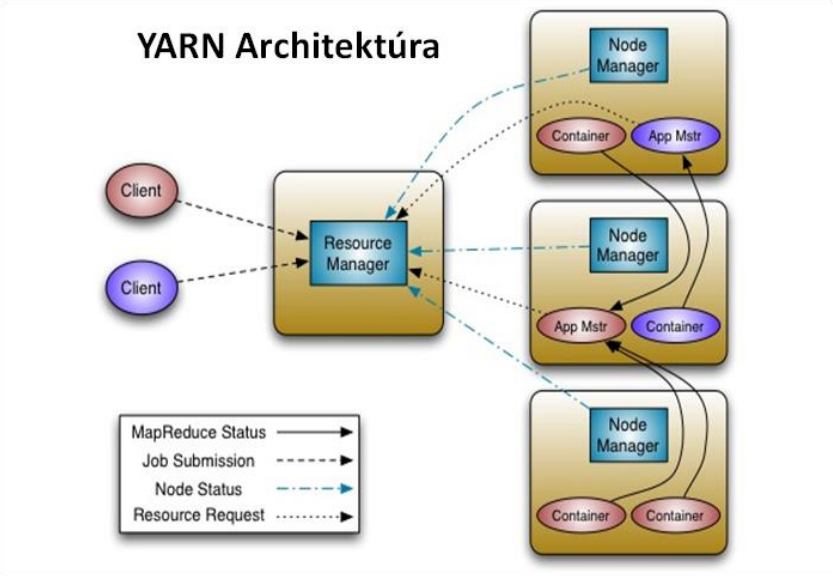
Yarn

S vyššou adaptáciou Hadoop a so zvyšujúcimi sa požiadavkami na jednotlivé klastre *(ktoré bežia niekoľko desiatok tisíc uzlov – nodes)* vznikla potreba na dosiahnutie lepšej synchronizácie, rozloženia zdrojov a ich optimalizácie. To boli hlavné dôvody vzniku YARN *(Yet Another Resource Negotiator)*, ktorý je podprojektom Apache Hadoop Project. Jeho umiestnenie v rámci Hadoop ecosystem je znázornené na obrázku.



Obrázok zdroj (edureka)

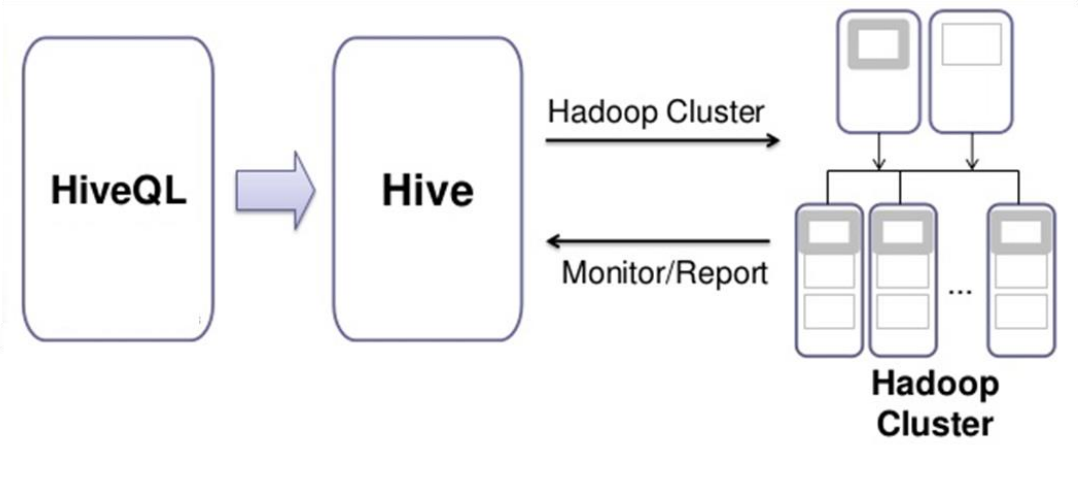
Architektúra Yarn sa skladá z dvoch hlavných komponentov. Prvým je ResourceManager /Scheduler, ktorý je zodpovedný za alokáciu zdrojov na jednotlivé bežiace aplikácie pričom, ale nevykonáva žiadny monitoring alebo záznam o stave jednotlivých aplikácií. Takisto nie je zodpovedný ani za prípadne reštarty aplikácií z dôvodu zlyhania aplikácie alebo hardwaru. Podporuje hierarchické zoraďovanie čo mu dovoľuje prediktívnejšie rozdelenie zdrojov klastra. Druhým komponentom je ApplicationsManager, ktorý je zodpovedný za akceptáciu jednotlivých úloh *(jobs)* na spracovanie, monitoruje ich priebeh a je zodpovedný za prípadný reštart ApplicationMaster *(zodpovedá koordináciu jednotlivých schránok/containers pridelených Scheduler-om)*. Dôležitým prvkom je aj NodeManager, ktorý reportuje Scheduler stav jednotlivých zdrojov *(ich CPU, kapacitu, sieť atď.)*. Popis YARN architektúry je znázornený na obrázku.



Zdroj Apache Foundation - Apache Hadoop NextGen MapReduce

Ďalšie aplikácie v rámci **Hadoop ecosystému** podporované **Apache Foundation** .

Hive - Apache Hive je Data Warehouse podporujúci analýzy dát, sumarizácie a vyhľadávanie dát uložených v Hadoop HDFS *(je kompatibilný so systémami ako Amazon S3)* . Jazykom Hive je HiveQL, ktorý je podobný štruktúrovaným jazykom SQL a zároveň podporuje MapReduce. Na zrýchlenie dotazovania používa indexovanie. Na obrázku je znázornené umiestnenie a interakcia Hive v rámci Hadoop ecosystému.



HBase open source databáze založená na nerelačnom dátovom modeli vytvorená po vzore Google Big Table beží na HDFS a poskytuje Big Table vybavenie Hadoop. Podrobnejšie popísaná v kapitole NoSQL.

Pig je platforma určená na analýzu veľkých objemov dát. Skladá sa zo skriptovacieho jazyka (Pig Latin) a prostredia na vytváranie programov MapReduce na Hadoop klastri. Oproti HiveQL je Pig flexibilnejším jazykom na formátovanie dát, má pomerne bohatý syntax a podporuje aj operácie ako:

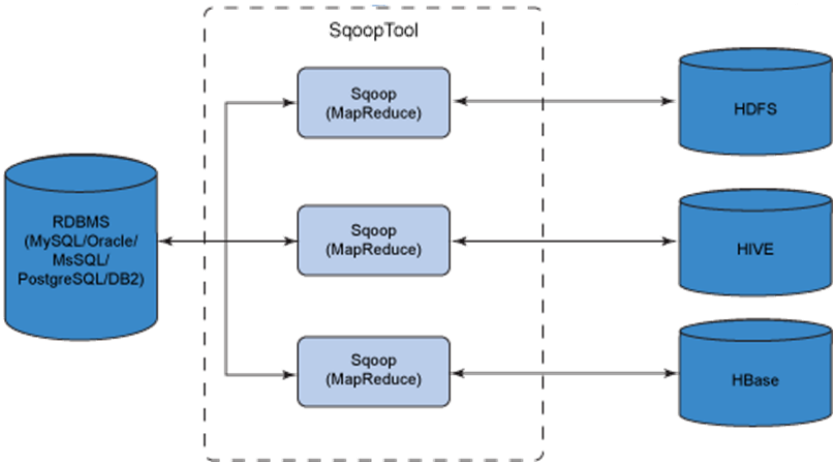
- Načítanie a ukladanie,
- Filtrovanie,
- Zgrupovanie a pridávanie,
- Triedenie,
- Stream-ovanie,
- Spájanie a delenie dát

Oozie je koordinátorom jednotlivých úloh a ich pracovného toku v rámci Hadoop, pričom zahŕňa do svojej kompetencie aj non-MapReduce úlohy. Je plne integrovaný v rámci Apache Hadoop ecosystému podporuje Apache Pig, Apache Hive, and Apache Sqoop. Zároveň sa môže použiť aj na niektoré špecifické typy úloh ako shell scripting alebo programy v Java.

Sqoop ("SQL-to-Hadoop") je nástroj na transfer dát medzi RDBMS a HDFS (oboma smermi) alebo inými Hadoop dátovými úložiskami napr. Hive alebo HBase.

Sqoop Architektúra

„toku dát“



ZooKeeper je služba na administráciu a uchovávanie informácií o konfigurácii systému, registra mien, poskytuje synchronizačné služby a celkovú koordináciu jednotlivých procesov. Zookeeper je distribuovaný systém uzlov *(Master- Slave)*.

Flume je nástroj na zber, agregovanie a nakladanie s veľkými objemy log-dát smerom do a von z Hadoop HDFS. Má jednoduchú a robustnú flexibilnú architektúru, ktorá je odolná na prípadne chyby a dislokácie.

Chukwa projekt Hadoop, ktorý sa venuje zberu a analýzam rozsiahlym log súborov. Je postavená na HDFS a MapReduce.

Návod na inštaláciu Hadoop *(iba v anglickom jazyku/ctrl+klik)*

Machine Learning a Big Data

Zhromažďovanie a spracovanie veľkých objemov dát je jedna vec, ale získať z nich relevantné informácie je druhá. Big Data výrazným spôsobom mení aj nástroje prediktívnej analýzy a zároveň mení celkové vnímanie získavania informácií z dát a ich interpretácie. Vo všeobecnosti tradičnému výskumu dát dominoval prístup „pokus - omyl“, ale s rastúcimi objemami dát a ich nehomogénnou štruktúrou je jeho požitie veľmi limitované. Navyše tradičné štatistické metódy sa sústreďujú na najmä statické analýzy dát *(časovo ohraničených dát)*. Z toho dôvodu je veľmi dobrou alternatívou, ktorá eliminuje spomenuté nedostatky Machine Learning.

Machine Learning je oblasť počítačovej vedy a umelej inteligencie *(AI)*, ktorá pojednáva o systémoch, ktoré sú schopné učiť sa z dát, ako len explicitne nasledovať programové zadanie. Okrem PV *(počítačovej vedy)* a AI je Machine Learning silne previazaný aj so štatistikou, štúdiom algoritmov a optimalizáciou systémov. Cieľom je poskytnúť čo najpresnejšie predikcie rôznych druhov a pre rôzne účely napr. odhaľovanie podvodov, produktové odporúčania, rôzne druhy segmentácií, rozoznávanie hovoreného slova alebo voľne písaných textov a pod.. Pričom dôraz je kladený na prediktívne analýzy v reálnom čase a vysokú prispôsobivosť systému.

Samotný Machine Learning z pohľadu významu pre Big Data by sme mohli rozdeliť podľa typu učenia na:

1. učenie s učiteľom *(supervised learning)*
2. učenie bez učiteľa *(unsupervised learning)*
3. učenie posilňovaním *(reinforcment learning)*
4. učenie s učiteľom aj bez *(semisupervised learning)*
5. multiúlohové učenie *(learning to learn/multitasking learning)*

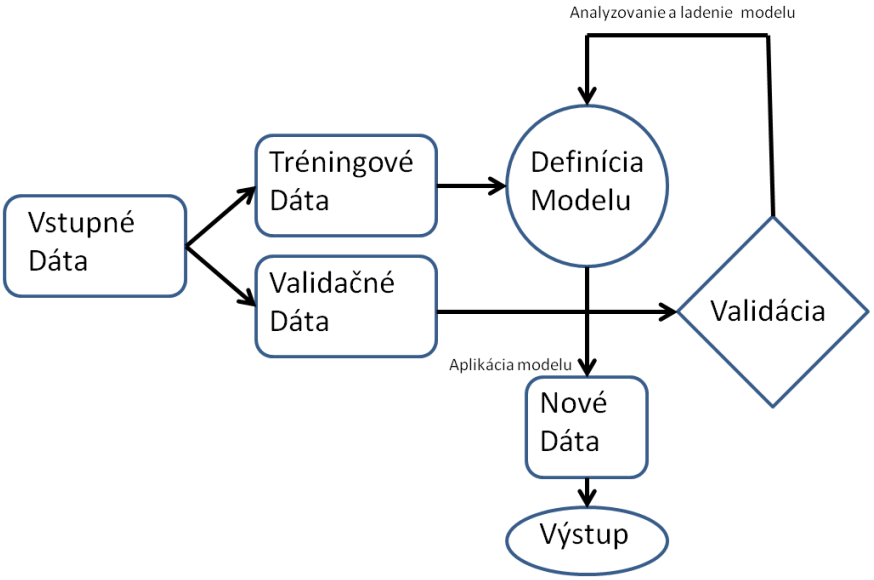
Učenie s učiteľom *(Supervised learning)*

Je metóda Machine Learning , ktorá pre učiacu sa funkciu používa tréningové dátové súbory. Tréningové dáta sa skladajú zo vstupných dát a požadovaného výstupu. Kontrolovaný učiaci sa algoritmus analyzuje tréningové dáta a odvádza z nich funkciu, ktorá môže byť použitá pri hodnotení nových dátových súborov rovnakého alebo podobného charakteru. Výstupom funkcie môže byť výsledná hodnota *(pri použití regresných modelov)* alebo môže predpovedať triedne/typové označenie vstupných dát *(pri použití klasifikačných modelov)*.

Samotný postup definície funkcie za pomoci učenia sa s učiteľom môžeme zadefinovať v pár jednoduchých krokoch *(viď. obrázok)* :

1. Príprava vstupných dát. Stanovenie typu dátového súboru.
2. Vytvorenie tréningového a validačného dátového súboru. Súbory môžu byť vytvorené náhodným rozdelením *(štandardné rozdelenie je tréningové dáta 70-80% a validačné 30-20%)* . Tréningové dáta používa model na učenia sa, a samotnú svoju definíciu tak, aby obsahoval dostatočný počet prvkov popisujúci požadovaný výstup a zároveň vhodných na použitie pri predikcii na ďalších dátových súboroch. Validačný dátový súbor sa používa na overovanie nadefinovaného modelu *(nadefinovaného pomocou tréningových dát)*, určenie

jeho chybovosti a presnosti. Celý cyklus sa opakuje až kým nedosiahneme z nášho pohľadu najoptimálnejší model.



3. Použitie nadefinovaného modelu na nových dátach.
4. Výstup.

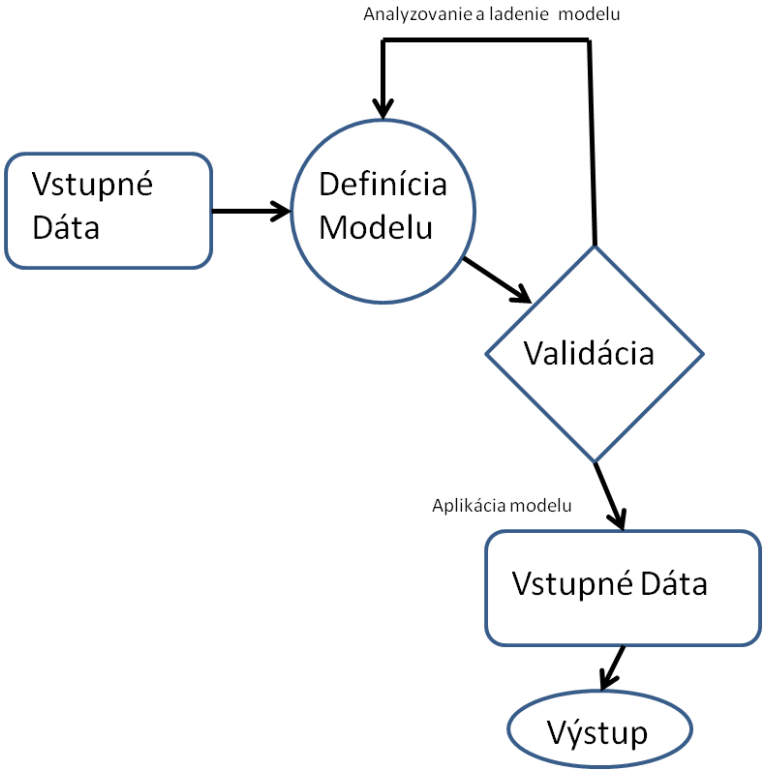
Učenie bez učiteľa *(Unsupervised learning)*

Zásadným rozdielom medzi učením sa bez učiteľa a s učiteľom je, že pri učení bez sa snažíme identifikovať skryté charakteristiky. To znamená, že modelu neposkytneme prvky popisujúce požadovaný výstup. Pri riadenom učení definujeme jednotlivé prvky tak, aby sme v čo najväčšej miere znížili jeho chybovosť. Pri neriadenom učení sa algoritmus snaží vytvoriť klastre s podobným obsahom a charakteristikami. V niektorých prípadoch nemusia jednotlivé vzťahy v rámci klastra byť jednoznačné na prvý pohľad, ale algoritmus sa priebežne snaží maximalizovať ich podobnosť v klastri a zároveň zvyšovať rozdiely medzi jednotlivými klastrami.
Ďalším výrazným rozdielom medzi riadeným a neriadeným učením je koncept Tréningového súboru dát *(tak ako bol spomenutý pri „učení sa s učiteľom)*, ktorý v pri neriadenom učení stráca význam. *Príklad: V prípade súboru demografických dát je prípadný nadefinovaný model učením sa bez učiteľa nepraktický, z dôvodu, že pri novom dátovom súbore by algoritmus hľadal nove väzby a charakteristiky v ňom a mohol vytvárať nové typy klastrov.*

Samotný postup definície funkcie za pomoci učenia sa bez učiteľa môžeme zadefinovať v pár jednoduchých krokoch *(viď. obrázok)* :

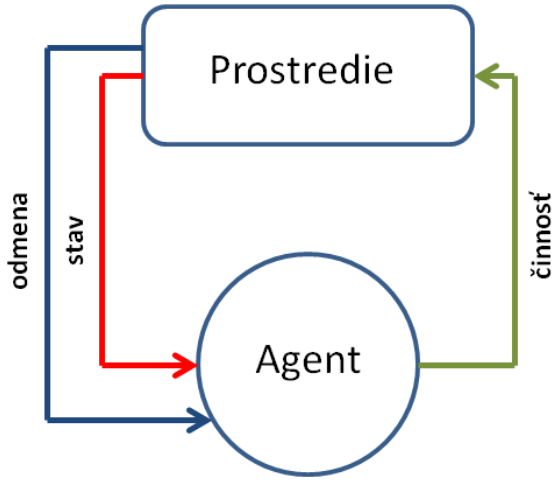
1. Príprava vstupných dát. Stanovenie typu dátového súboru.
2. Definícia modelu: Algoritmus beží na dátovom súbore s cieľom vytvoriť skupiny/klastre so vzájomnými väzbami a rovnakými alebo podobnými charakteristikami.
3. Validácia: Jednotlivé klastre potrebujeme následne verifikovať sériou štatistických úkonov, aby sme znížili prípadnú chybovosť.

4. Výstup.



Učenie posilňovaním *(Reinforcment learning)*

Je oblasť Machine Learning inšpirovaná behaviourálnou psychológiou. Zaoberá sa výkonom činností softwarového agenta v definovanom prostredí tak aby sa maximalizovala jeho odmena. *Príklad: Pes získava črty správania učením sa na základe odmeny a trestu, pričom sa snaží maximalizovať odmenu a minimalizovať trest. Softwarový model funguje na tom istom princípe.*
Jednoduchá schéma učenia sa posilňovaním je znázornená na obrázku:

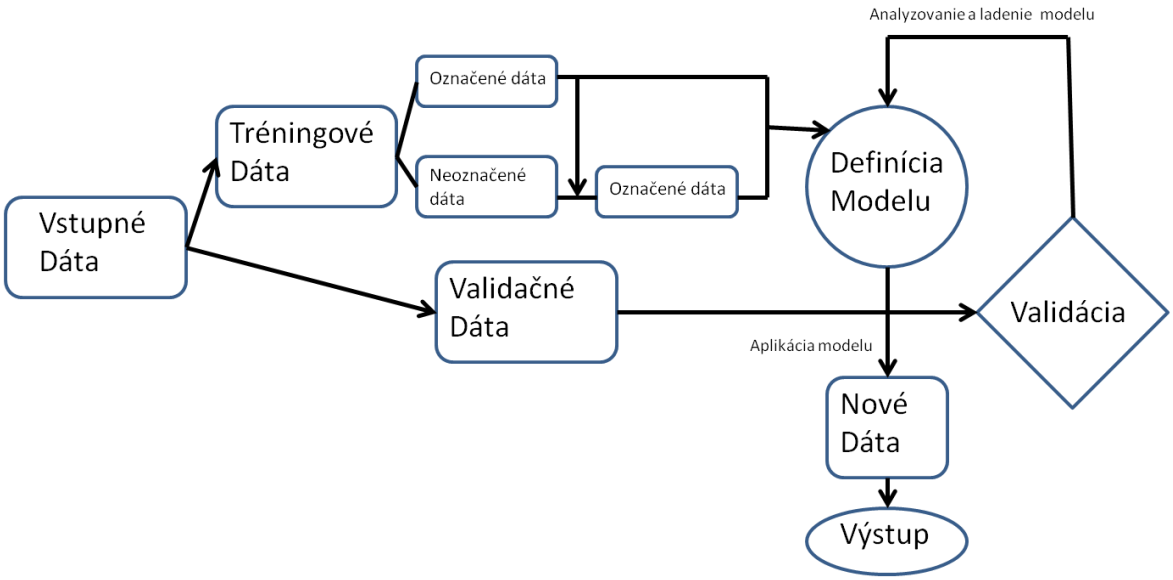


Agent je v priamej interakcii s prostredím, pričom dochádza ku zmenám prostredia a k odmene za jeho činnosť. Pozitívna alebo negatívna odmena je informáciou, ktorá definuje jeho činnosť do budúcna,

reakciu v podobnom prostredí, situácií. Cieľom je maximalizovať pozitívnu odozvu. Interakcia agenta s prostredím je modelovaná ako Markov Rozhodovací Proces *(Poskytuje matematický rámec pre modelové rozhodovanie v situáciách, kde výsledky sú z časti náhodné a z časti pod kontrolou užívateľa)*.

Učenie s učiteľom aj bez *(Semisupervised learning)*

Je hybridom učenia s učiteľom *(supervised learning)* a učenia bez učiteľa *(unsupervised learning).* Keď by sme porovnali učenie s učiteľom a učenie bez učiteľa tak v prvom prípade poznáme požadovaný výstup *(môžeme ho definovať ako označené dáta)* zatiaľ čo v druhom výstup je neznámy *(môžeme ho definovať ako neoznačené dáta).* Kombinácia označených a neoznačených dát v jednom modeli je učenie s učiteľom aj bez *(semisupervised learning).*
Jeho princíp a rozdiel oproti klasickému učeniu sa s učiteľom je vysvetlený na schematickom obrázku.



Vo všeobecnosti sa na definíciu modelu používa malé množstvo označených dát s väčším množstvom neoznačených. Takýto postup môže zvýšiť presnosť modelu a znížiť jeho chybovosť oproti učeniu bez učiteľa.

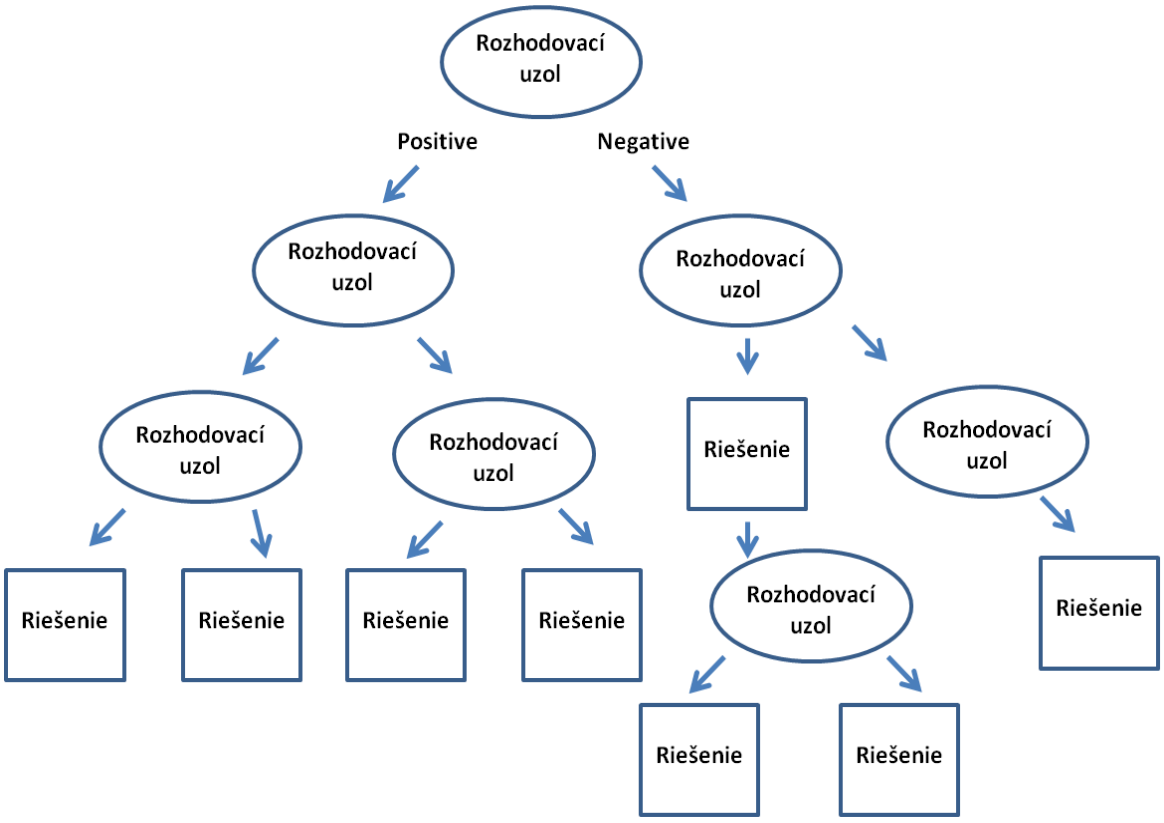
Multiúlohové učenie *(Learning to learn/multitasking learning)*

Pri multiúlohovom učení sa model učí pri práci s jedným problémom zároveň pracovať aj s problémami ,ktoré s ním priamo súvisia. Čo vedie k lepšej definícií konečného modelu pre riešenie hlavnej úlohy. Veľmi dobrým príkladom multiúlohového učenia je spracovanie spam-u *(nevyžiadanej pošty)*. Každý má iné podmienky na definíciu spam-u napr. pre niekoho môže byť spam všetko čo obsahuje ako krajinu pôvodu Nigériu alebo obsahuje údaje priamo súvisiace s touto krajinou, zatiaľ čo samotných Nigérijčanov to by ako definícia spam-u bolo nevyhovujúce. V tomto ohľade pracuje multiúlohové učenie s dedukciou a definuje optimálne nastavenie služby spam.

Z pohľadu používaných modelov by sme mohli Machine Learning rozdeliť *(uvedené sú len niektoré modely – čiastkové členenie so stručným popisom, obsiahlejší zoznam môže byť videný napríklad na stránkach wikipédie)*:

Rozhodovacie stromy *(Decision tree learning)*

Rozhodovacie stromy sú klasifikátor so stromovou štruktúrou a používajú na predikcie mapovanie jednotlivých pozorovaní objektov. Schéma rozhodovacieho stromu je znázornená na obrázku:



Rozhodovací strom pri riešení úlohy začne v koreni stromu a prechádza cez jednotlivé uzly až k listu. Vnútorné uzly sa nazývajú rozhodovacie, kde sledovaný objekt alebo úloha musí byť vyjadriteľná pomocou pravidiel s fixnou množinou vlastností, atribútov tak, aby každý možný výsledok testu bol reprezentovaný jednou vetvou. Konečný list stromu indikuje výslednú hodnotu – riešenie. Sú veľmi vhodným nástrojom tam kde potrebujeme jasnú indikáciu, ktoré oblasti sú najdôležitejšie pre predikciu alebo klasifikáciu a zároveň sú nenáročné na pochopenie. Nie sú veľmi vhodné na predikciu hodnôt kontinuálneho charakteru a môže sa u nich zvyšovať chybovosť pri vysokom počte atribútov v prípade, že máme malý počet tréningových príkladov.

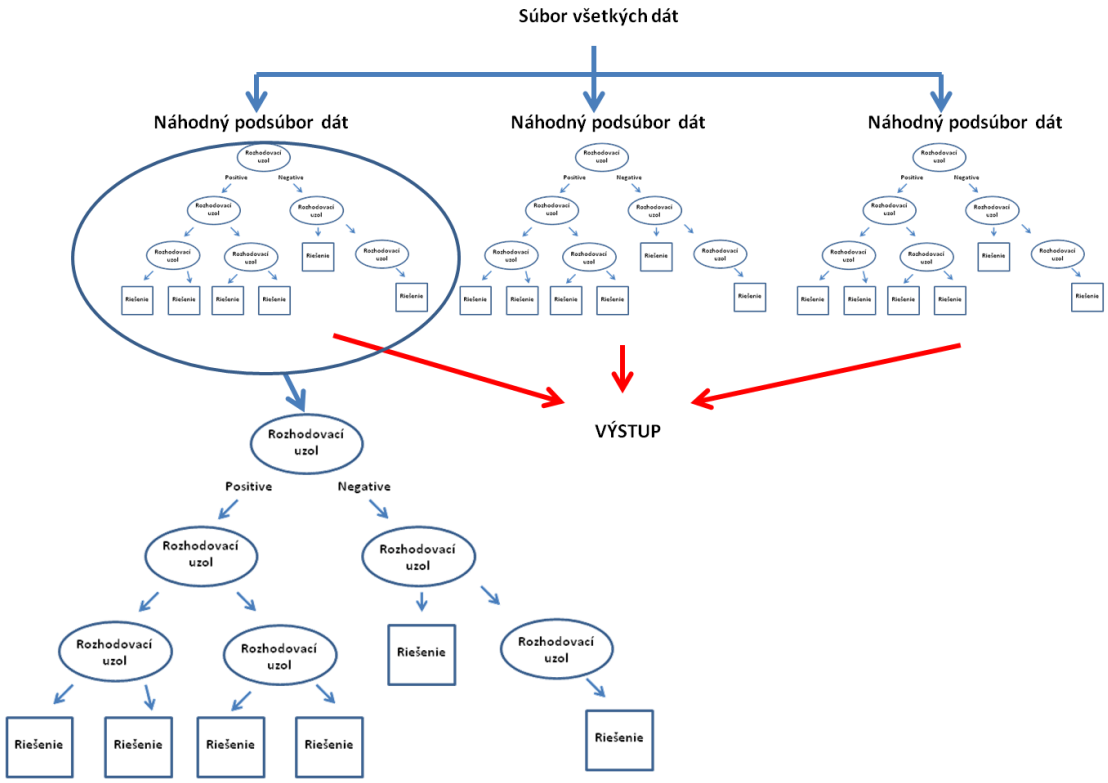
Rozhodovacie stromy použité v Big Data môžeme rozdeliť na 2 základné typy:

- Klasifikačné stromy – kde výstupom je trieda ku ktorej sledované dáta prináležia
- Regresné stromy – kde výslednou hodnotou je reálne číslo

Veľmi rozšírenou a populárnou metódou pri Big Dáta analýzach a predikciách je Classification And Regression Tree *(CART)* , ktorá používa obidva typy rozhodovacích stromov.

Random Forest

Je metóda skupinového učenia pre klasifikáciu *(a regresiu)* , ktorá funguje na princípe tvorby niekoľkých rozhodovacích stromov v rovnakom čase a s výstupom, ktorý je definovaný výstupmi jednotlivých stromov. Schematický popis je znázornený na obrázku.

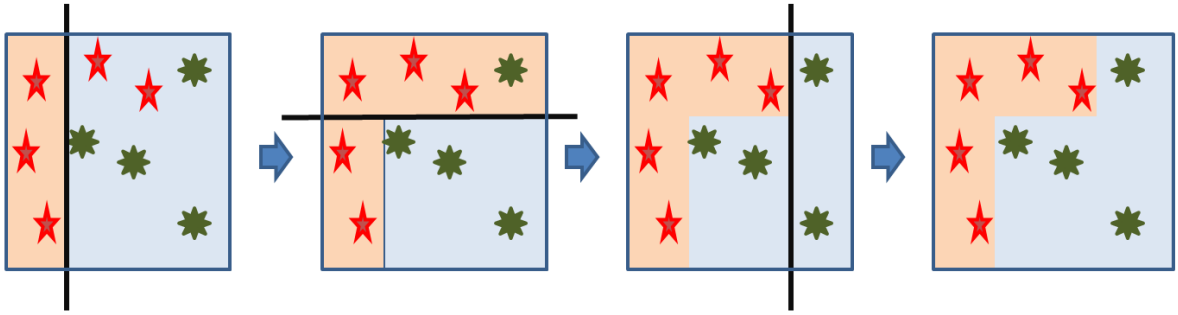


Random Forest veľmi dobre pracuje s veľkými objemami dát a zároveň dokáže zvládnuť tisíce vstupných premenných bez vymazania akejkoľvek z nich. Je to efektívna metóda pre odhad chýbajúcich údajov a udržuje presnosť ak chýba veľká časť údajov.

Boosting

Boosting je Machine Learning algoritmus na zníženie zaujatosti/odchýlky pri učení sa s učiteľom. Prístup, ktorý je založený na myšlienke vytvorenia vysoko prediktívneho klasifikátora na základe spojenia väčšieho množstva slabších a menej presných klasifikátorov.

Prvým krokom je vytvorenie klasifikátora s presnosťou vyššou ako 50% následne sú pridávané ďalšie klasifikátori majúce rovnaké klasifikačné vlastnosti. Výsledkom je vygenerovaný klasifikátor. Jednoduchý schematický postup tvorby Boosting klasifikátora je znázornený na obrázku.



1.Krok - vytvorenie prvotného klasifikátora

2.Krok – pridanie klasifikátora s rovnakými klasifikačnými vlastnosťami

3.Krok – ladenie s cieľom zvýšenia výkonu algoritmu

Výsledok

Veľmi populárnym a aj najdôležitejším Boosting algoritmom je AdaBoost *(Adaptive Boosting)*, často používaný aj s inými algoritmami na zvýšenie ich výkonu. AdaBoost umožňuje pri návrhu pridávať slabšie klasifikátori tak dlho pokiaľ nie je dosiahnutá požadovaná hodnota chybovosti skupiny klasifikátorov. Z pohľadu tvorby tréningového súboru je nutné spomenúť, že každá tréningová vzorka dostáva pridelenú váhu, ktorá definuje jej pravdepodobnosť výberu do tréningovej podmnožiny pre jednotlivé klasifikátori z vytváranej skupiny. V prípade, že je vzorka klasifikovaná presne jej šance na opätovný výber pre test na ďalšom klasifikátore klesajú a vice versa. AdaBoost sa takto sústredí na náročnejšie vzorky z tréningového súboru. Prvotné pridelenie váh pre všetky vzorky je jednotné a mení sa v každom iteračnom kroku.

Analýzy klastra *(Cluster Analysis)*

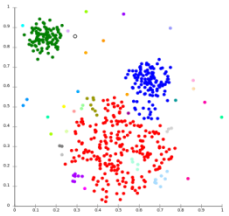
Je zlučovanie objektov do skupín *(klastrov)* tak, aby jednotlivé objekty v rámci skupiny boli rovnaké alebo v najväčšej možnej miere podobné. Zároveň jednotlivé skupiny mali jednoznačne odlišné charakteristiky. Príkladom analýzy klastra je napríklad aj analýza frekvencie slov v texte, vizualizácia takej analýzy je znázornená na obrázku.

Uvedený príklad je len jedným typom analýzy klastra. Existuje niekoľko rôznych algoritmov , ktoré sa pomerne líšia v definícií skladby objektov klastra a v účinnosti ich nájdenia a väzieb medzi nimi.

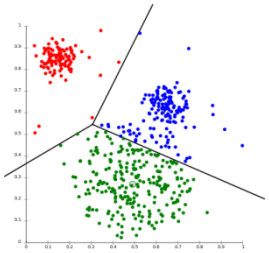
Vo všeobecnosti by sme zoskupovanie *(clustering)* mohli formulovať ako multi - objektovú optimalizáciu. Samotná analýza klastra nie je jednoprocesová samočinná úloha, ale jedná sa o opakovaný proces hľadania a multi - objektovej optimalizácie systémom úspech – neúspech. Pričom v priebehu samotnej optimalizácie bude niekedy potrebné modifikovať sledované parametre s cieľom čo najviac sa priblížiť požadovanému výsledku.

Typické modely analýzy klastra sú:

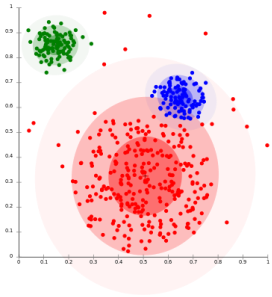
- **Model konektivity** – hierarchický model založený na vzdialenostiach sledovaných objektov



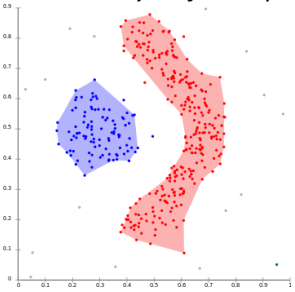
- **Ťažiskový model** – model založený na určovaní stredu/ťažiska a následne zoskupovanie okolo neho *(najznámejším príkladom je k-means)*



- **Distribučný model** – model s použitím štatistickej distribúcie



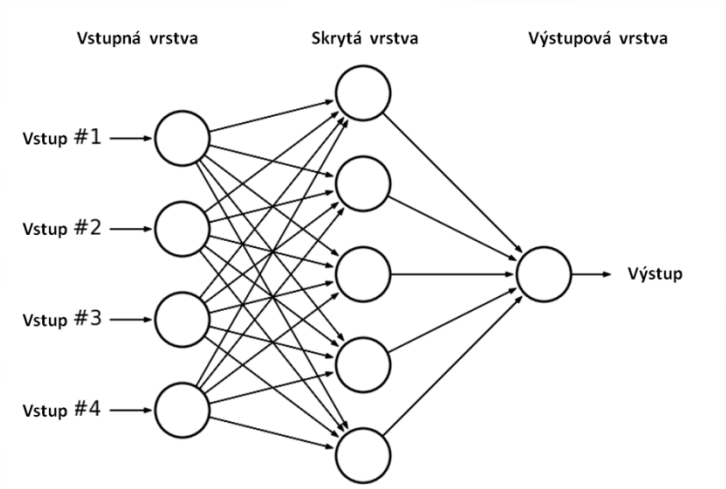
- **Model intenzity/hustoty** – je definovaný ako oblasti s najväčšou hustotou *(oblasti s väčšou hustotou dát ako je hustota vo zvyšnej časti priestoru)*



- **Dvojrežimový model** – dovoľuje simultánne zoskupovanie stĺpcov a riadkov v matrixe

Neurónová sieť *(Neural Network)*

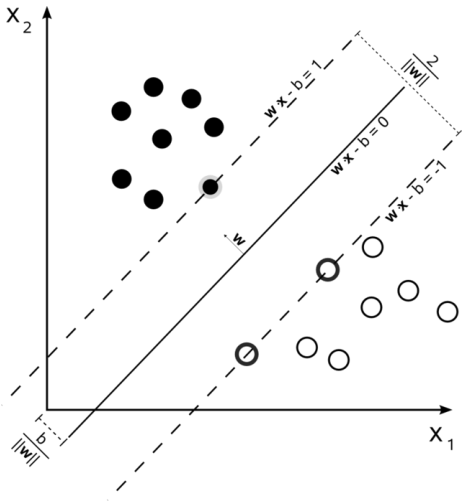
Je model, zostavený na základe abstrakcie vlastností biologických nervových systémov. Neurónová sieť má schopnosť uchovávania informácií a umožňuje ich ďalšie spracovanie, pričom napodobňuje ľudský mozog v zbieraní poznatkov v procese učenia a uchovávaní týchto poznatkov s využitím medzineurónových spojení *(skrytá vrstva)*. Schematické znázornenie neurónovej siete je uvedené na obrázku:



Základným prvkom neurónovej siete je neurón. Vo všeobecnosti má niekoľko vstupov od iných neurónov alebo z okolitého prostredia. Neurón transformuje svoje vstupy na výstup pomerne jednoduchou operáciou, zložitosť neurónovej siete spočíva v spojení mnohých takýchto jednoduchých elementov do celku. V prevažnej väčšine sietí sú neuróny usporiadané do vrstiev *(vstupnej, skrytej a výstupnej)*. Samotná informácia sa šíri od vstupných neurónov *(neuróny, ktorých vstupmi sú signály z prostredia)* cez skryté neuróny *(neuróny, ktoré sú vstupmi aj výstupmi spojené s inými neurónmi; tieto sa v niektorých typoch sieti nemusia vôbec nachádzať)* k výstupným neurónom *(neuróny, ktorých výstup vedie do prostredia).*

Support Vector Machines *(SVM)*

SVM je učenie s učiteľom, ktoré analyzuje dáta a rozoznáva v nich jednoznačné charakteristiky, ktoré sa používajú pre klasifikáciu a regresnú analýzu. Prvotné tréningové dáta sú označené ako patriace do jednej z dvoch uvedených kategórií, následne SVM učiaci sa algoritmus vytvára model, ktorý priraďuje nové dáta do jednej alebo druhej kategórie, čo môžeme nazvať ako binárny lineárny klasifikátor. SVM znázorňuje jednotlivé dáta ako body v priestore, mapované tak, aby jednotlivé kategórie boli jasne rozdelené *(v takej šírke ako je to maximálne možné)*.



(zdroj)

Aplikácie Machine Learning v Big Data prostredí

Apache Mahout

je open source projekt podporovaný Apache Foundation, ktorého cieľom je vytvoriť škálovateľnú Machine Learning knižnicu.

Kde pod slovom „škálovateľnosť“ je myslené:

1. Schopnosť spracovávať veľké objemy dát. V súčasnosti Mahout algoritmy, ktoré sú klastering *(klastrové analýzy),*klasifikácia a filtrovanie *(collaborative filtering)* sú implementované na vrchu distribučného systému Hadoop,
2. Nastaviteľná podpora podľa potrieb firiem alebo projektov. Pričom Mahout je distribuovaný aj pod komerčnou licenciou Apache Software license..

Príklady použitia Mahout :

Odporúčania založené na sledovaní správania užívateľa.

Zoskupovanie *(klastrové analýzy)* napr. textových dokumentov podľa tém.

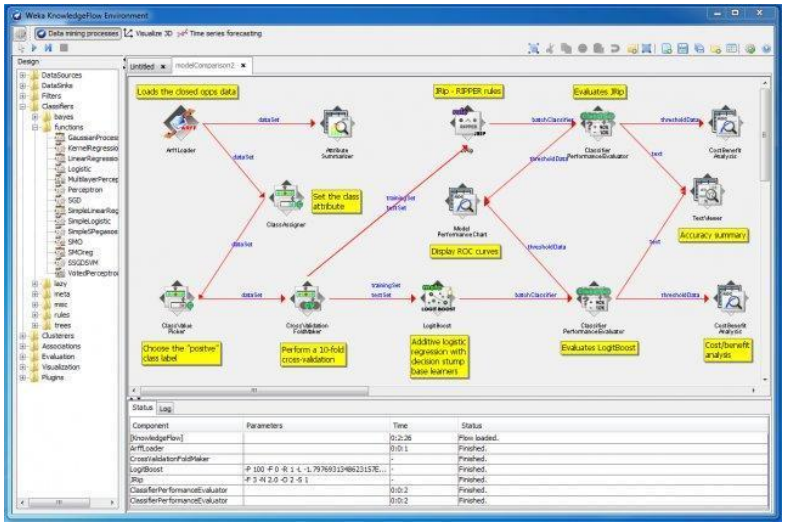
Download Mahout

Weka

Waikato Environment for Knowledge Analysis, je pomerne populárny voľne dostupný nástroj Machine Learning, vyvinutý na univerzite Waikato. Obsahuje sadu vizualizačných nástrojov a algoritmov slúžiacich k dátovej analýze a predikcii, pričom poskytuje grafické rozhranie (*vid. obrázok)*. Je plne implementovaná v prostredí Java a obsahuje techniky na predspracovanie dát, klasifikáciu, regresné analýzy a klasterové analýzy. Nevýhodou je, že vyžaduje, aby dátové body boli popísané pevným počtom atribútov.

Weka obsahuje uvedené modely :

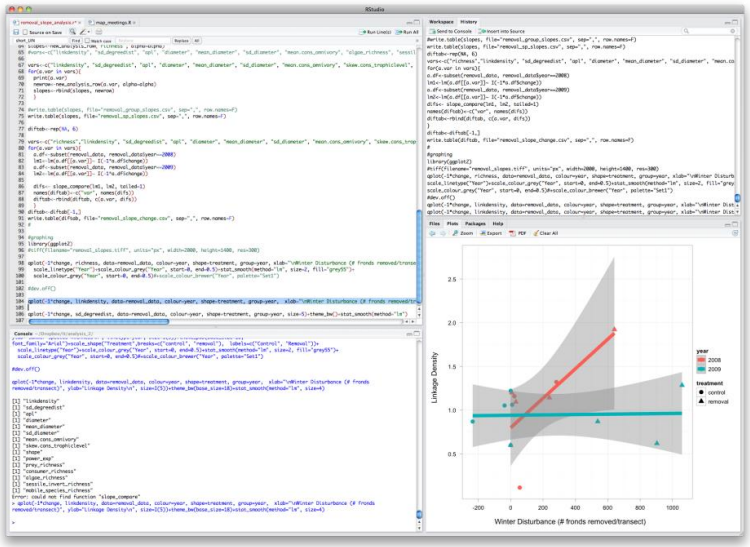
- rozhodovacie stromy
- učenie sa s učiteľom
- support vector machines
- lokálne váženú regresiu
- analýzy klastra



Návod na inštaláciu Weka

R

R je voľne dostupný programovací jazyk a softwarové prostredie pre štatistické výpočty a grafickú vizualizáciu. V posledných rokoch jeho popularita prudko narastá aj vďaka dobrému uplatneniu v Machine Learning. R softwarové prostredie je napísané v C, Fortran-e a R a je voľne dostupné pod GNU. Používa rozhranie príkazového riadku, ale veľmi populárne je aj vylepšené užívateľské rozhranie RStudio *(vid. obrázok).* Jeho výhodou oproti softvérovým balíkom ako napríklad Microsoft Excel, GenStat a Statistica je, že tie napriek výborným nástrojom na analýzu a vizualizáciu, nie sú až tak flexibilné na pridávanie nových funkcionalít, spolupráca s ďalšími nástrojmi nie je priamočiara a tieto nástroje priamo predpokladajú prítomnosť užívateľa, ktorý vykonáva osobne analýzy cez grafické ovládacie prostredie. Na druhej strane bežné programovacie jazyky neobsahujú zabudované neelementárne matematické funkcie a tie je nutné naprogramovať skoro od základov. A práve medzeru medzi týmito prístupmi sa snaží vypĺňať R.



Návod na inštaláciu R

scikit-learn

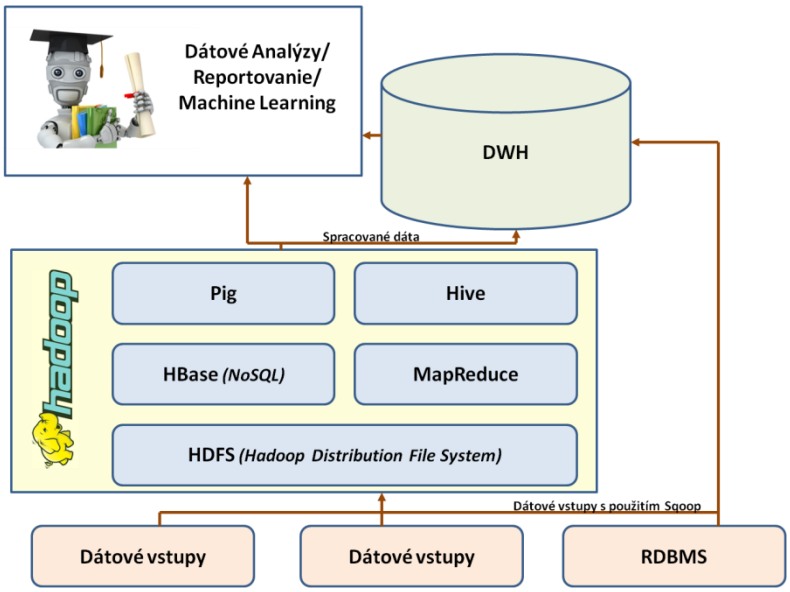
Je open source Machine Learning knižnica pre Python. Obsahuje rôzne typy klasifikácie ,regresnej analýzy, analýzy klastra ako aj SVM alebo Random Forest. Napisaná v jazyku Python s niektorými algoritmami napísanými v jazyku Cython, je postavená na numerickej a vedeckej kniznici NumPy and SciPy.

Je vhodné ešte spomenúť nástroje ako MATLAB, ktoré sú vhodné na rýchlu definíciu a prvotné testovanie Machine Learning algoritmov. Sústredia na modelovanie, návrhy algoritmov, simulácie, analýzy a prezentácie dát, paralelné výpočty a návrhy riadiacich a komunikačných systémov. Veľmi dobrou alternatívou k MATLAB je programovací jazyk Octave . Octave je voľne dostupný pod GNU a je kompatibilný s MATLAB.

Návrh prístupu k testovaniu Big Data architektúry v prostredí Štatistického úradu Slovenskej Republiky

Testovanie Big Data architektúry je značnou výzvou pre ŠÚSR najmä z dôvodu nedostatku znalostí v danej oblasti ako samotný test realizovať, aké dáta a v akých objemoch použiť na testovanie. Vo všeobecnosti organizácie čelia problémom s definíciou testovacích stratégií, nastavenia optimálneho testovacieho prostredia, validáciou štruktúrovaných a neštruktúrovaných dát, práce s NoSQL ako aj so samotnou realizáciou testovania. To spôsobuje zhoršenie dátových výstupov z testovania, odďaľuje samotnú implementáciu a zvyšuje náklady. Uvedený všeobecný návrh si všíma všetky uvedené výzvy a problémy pričom sa snaží poskytnúť riešenie založené najmä na voľne dostupných zdrojoch *(open source softwarových riešeniach).* Koncept testovania predstavený v tejto kapitole kladie dôraz na validáciu kvality dát v jednotlivých fázach procesu ich spracovania. Netestuje technologické alebo softwarové vybavenie.

Návrh jednoduchej všeobecnej architektúry pre účel testovania je znázornený na obrázku.



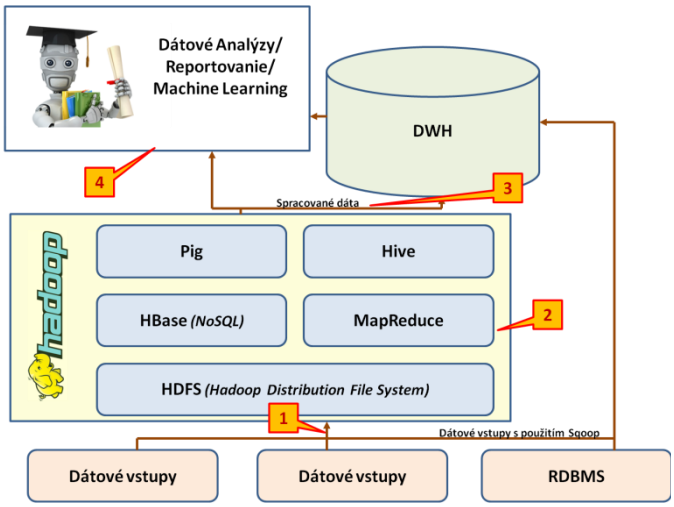
Jednotlivé prvky sú buď tradičné komponenty ako DWH ,RDMBS a tradičné metódy dátovej analýzy, alebo open source Hadoop *(NoSQL a MapReduce)* a Machine Learning, ktorým boli venované samostatné kapitoly. Dátové vstupy predstavujú menej tradičné zdroje súčasnej štatistiky a to údaje získané z internetu *(crawl jobs)* a sociálnych sietí. Sqoop je rozhranie *(rozhranie príkazového riadku),* ktoré slúži na transfer dát medzi Hadoop a RDBMS.

K samotnému testovaniu potrebujeme pomerne rozsiahly dátový set *(u ktorého predpokladáme určitú kvalitu – či už vo forme jeho nízkej chybovosti alebo relevantnosť vhodnú pre dátové analýzy)* a testovacie prostredie. Pričom testovaním sledujeme **funkčnosť** ako napríklad validáciu procesov MapReduce, validáciu štruktúrovaných a neštruktúrovaných dát. Okrem funkčnosti sledujeme a prispôsobovateľnosť procesov ako aj samotnú **výkonnosť** systému voči **základným charakteristikám** Big Data *(objem ,rôznorodosť a rýchlosť).*

Testovanie by malo prebiehať v každom štádiu sprocesovania dát v rámci Big Data systému. Testovanie zahŕňa:

1. Validácia dát pred Hadoop sprocesovaním
2. Validácia dátového výstupu Hadoop MapReduce procesu
3. Validácia dát pre extrakciou a uložením do DWH
4. Test Reportu

Jednotlivé fázy testovania sú znázornené na obrázku.



Validácia dát pred Hadoop sprocesovaním

Extrahované dáta z rôznych zdrojov na základe vopred definovaných podmienok sú načítane do HDFS pred ďalším sprocesovaním.

Možné nástrahy:

Pri procese extrahovania dát hrozí nebezpečenstvo, že dáta nebudú správne načítané do Hadoop, môže dôjsť k ich nesprávnemu uloženiu alebo nesprávnej replikácii.

Validácia:

1. Porovnať extrahované dáta voči dátam priamo na zdroji či boli extrahované správne a nedošlo k zmene ich obsahu alebo významu
2. Overiť podmienky extrakcie či boli extrahované iba požadované dáta
3. Overiť správnosť načítania dát do HDFS
4. Potvrdiť ,že vstupné dáta boli prerozdelené a replikované do rôznych uzlov

Validácia dátového výstupu Hadoop MapReduce procesu

Keď sú dáta načítané do HDFS tak následne ich Hadoop MapReduce sprocesuje.

Možné nástrahy:

Počas tejto fázy sa môžu vyskytnúť problémy s interpretáciou bežiaceho programu. Pig a Hive sú pomerne komplexné programy, ktoré bežia v rámci Hadoop MapReduce na veľkých objemoch dát na rôznych uzloch. Hadoop poskytuje možnosť bežať distribuovaný proces spracovania na viacerých počítačových klastroch. MapReduce je rozdelený na množstvo menších operácií, kde každá môže byť vykonaná *(aj opätovne vykonaná)* na hociktorom uzly v rámci klastra.

Možným rizikom je, že jednotlivé činnosti nemusia bežať správne *(napríklad: jeden uzol môže bežať bezchybne, zatiaľ čo skupina uzlov môže vytvárať chybné agregácie).* Chyby môžu nastať v konfigurácii jednotlivých uzlov alebo chybnom formáte výstupu.

Validácia:

1. Overiť, že sprocesovanie dát je dôkladne ukončené a výstupný súbor bol vytvorený
2. Overiť MapReduce proces, že páry kľúč - hodnota *(key-value pair)* sú správne vygenerované
3. Overiť agregáciu a konsolidáciu dát po Reduce procese
4. Potvrdiť kompletné sporocesovanie dát, porovnaním vstupných a výstupných dát
5. Porovnať formát výstupných dát voči požiadavkám na vstupe

Validácia dát pred extrakciou a uložením do DWH

Po ukončení MapReduce a vygenerovaní výstupných súborov, dáta sú umiestnené do DWH *(alebo iného transakčného systému podľa požiadaviek)*.

Možné nástrahy:

Nesprávna aplikácia pravidiel transformácie, extrakcie z Hadoop HDFS a uloženie do DWH.

Validácia:

1. Overiť, že pravidlá transformaácie boli správne aplikované
2. Porovnať výstupné dáta *(dáta v DWH)* s dátami v HDFS či nedošlo k poškodeniu alebo zničeniu dát

Test Reportu

Reporty sú generované pomocou reportovacích nástrojov, ktoré získavajú dáta z DWH alebo pomocou dotazovania v Hive.

Možné nástrahy:

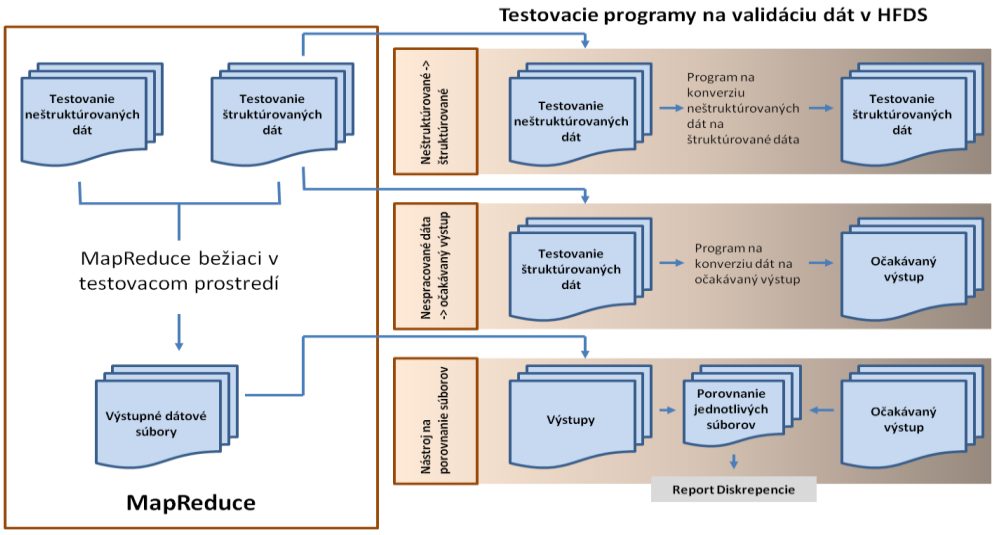
Počas generovania reportov môžu nastať problémy s definíciou alebo formátom reportov *(nespĺňajú požadované kritéria)* , špecifikáciou dát alebo vysokou dátovou chybovosťou.

Validácia:

1. ETL *(extract transform load)* beží pre všetky dátové zdroje počas umiestňovania dát do DWH. Počas tohto procesu je vhodné priebežne kontrolovať už umiestnené dáta či ich transformácia prebehla bezchybne a či dáta obsahujú všetky informácie, ktoré budú potrebné pri ďalšom spracovaní.
2. Cube testing – testujeme plnenie komplexných príkazov *(s viacerými atribútmi na riadky/ sĺpce/ atď.)*. Verifikujeme, že jednotlivé hierarchie v rámci dimenzií s pred - agregovanými hodnotami sú správne spočítane a znázornené v reporte-
3. Dashboard testing – sledujeme či nástoj a jeho jednotlivé prvky sú aktualizované

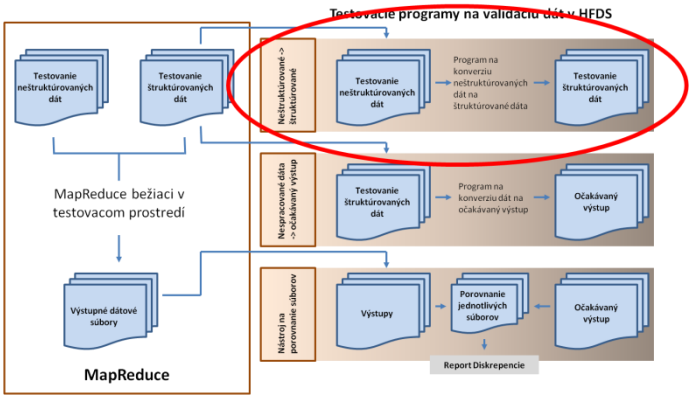
Ďalším aspektom na ktorý treba klásť dôraz pri testovaní sú základné charakteristiky Big Data objem *(volume),*rôznorodosť *(variety)* a rýchlosť *(velocity).*

Volume – veľké objemy dát prichádzajúce do systému musia byť spracované a analyzované, pričom problém vzniká pri verifikácií a validácií takýchto objemov. Manuálna validácia a verifikácia by bol extrémne zdĺhavý proces z toho dôvodu je nutné použiť porovnávacie programy. Dokonca aj pre porovnávacie programy to môže byť časovo náročný proces, preto sa snažíme bežať paralelne niekoľko takýchto programov na rôznych uzloch *(tak ako sú dáta priebežne spracované s MapReduce)* alebo porovnávame iba vybraté vzorky pričom kladieme dôraz na to, aby vzorky pokryli čo najviac možných scenárov . Na obrázku vidíme schematický znázornený proces porovnávania.



Dáta sú konvertované do formátu očakávaného výstupu a následne porovnávané pomocou softwarových nástrojov. Nevýhodou tohto spôsobu testovania je, že si vyžaduje viac času na prípravu *(najmä na programovanie).*

Variety – Neštruktúrované dáta nemajú formát preto ich testovanie *(validácia)* je pomerne náročný proces do komplexnosti aj časovo. Automatizácia tohto procesu sa môže čiastočne dosiahnuť konvertovaním dát na štruktúrované pomocou jazyka Pig *(viď. obrázok).*



Dosiahnuť automatizáciu pre celý neštruktúrovaný dátový súbor je veľmi náročné z dôvodu ťažko predvídateľného správania sa dát *(vstupné dáta môžu mať rôzne formáty, ktorý sa môže meniť pri každom novom teste).* Z toho dôvodu je potrebné definovať rôzne testovacie scenáre pre požadovaný cieľový výstup voči ktorým sú dáta následne testované *(pričom je možné realizovať testy na dátových vzorkách).* Semi - štruktúrované dáta tiež nemajú formát, ale ich štruktúru je možné vydedukovať na základe podobných charakteristík ktoré obsahujú. Pre validáciu musia byť v prvom kroku transformované na štruktúrovaný formát pomocou identifikácie ich spoločných charakteristík. Samotná validácia prebieha pomocou porovnávacích softwarových nástrojov.

Velocity – Rýchlosť akou sú nové dáta tvorené je jednou zo základných charakteristík Big Data. Z toho dôvodu je potrebné identifikovať či testovaní systém je schopný ich zvládať a to sériou záťažových testov. Jednotlivé testy prebiehajú pod rôznou rýchlosťou prúdu dát a ich načítavania do systému.

Testovanie výkonnosti – ma kľúčovú úlohu v každom Big Data projekte. V prípade zlej architektúry alebo chybne napísaného programu výkonnosť celého systému výrazne klesá. Oblasti kde sa môžu zjaviť problémy s výkonnosťou sú napríklad počas MapReduce procesu nerovnováhy pri delení vstupných dát, triedenie alebo agregácia dát. Test výkonnosti je vykonaný v prostredí, ktoré je rovnaké

alebo veľmi podobné reálnemu prostrediu a s veľkými objemami dát. Jednotlivé metriky výkonnosti sú následne zachytené pomocou Hadoop monitorovacieho nástroja *(Hadoop performance monitoring tool).* Príkladom takýchto metrík sú napríklad čas trvania alebo využitie pamäte.
Dôležitou súčasťou testovania je testovanie prevzatia služieb pri zlyhaní (Failover Testing). Hadoop architektúra sa skladá z uzla *(name node)* a stoviek dátových zápisov *(data notes)* uložených na navzájom prepojených serveroch. Z toho dôvodu existuje nebezpečenstvo, že v prípade zlyhania uzla alebo siete sa niektoré komponenty HDFS môžu stať nefunkčné. Architektúra Hadoop je navrhnutá s detekciou porúch, tak aby sa automaticky obnovila a pokračovala v spracovaní dát. Pri testovaní je potrebné sa sústrediť na proces obnovenia funkčnosti a zabezpečenia nerušeného procesu spracovania dát automatickým prepnutím z chybného uzla na funkčný. Priebeh testu kontrolujeme pomocou súborov kontrolných bodov *(ktoré obsahujú podrobný zápis o priebehu procesu)*. Medzi hlavné sledované metriky patria čas obnovenia alebo maximálna doba prerušenia prevádzky systému.

Testovacie prostredie – vytvorenie vhodného testovacieho prostredia je pravdepodobne najväčšia výzva procesu testovania. Vytvorením testovacieho prostredia v cloud-e je možné získať flexibilitu v prístupe, údržby a optimalizácií testovacieho prostredia. Základné kroky nastavenia testovacieho prostredia:

1. Posúdenie Big Data architektúry
    a. Posúdenie požiadaviek proces spracovania dát
    b. Určenie vhodného počtu uzlov
    c. Definícia požiadaviek na Cloud *(bezpečnosť, veľkosť atď.)*
    d. Posúdenie softwarových požiadaviek *(Hadoop,NoSQL atď)*
2. Návrh testovacieho prostredia
    a. Cloud test infraštruktúra *(požiadavka každého uzla na RAM, veľkosť disku atď)*
    b. Určenie poskytovateľa cloud služieb
    c. Definícia SLA
    d. Definícia testovacej stratégie
3. Implementácia a údržba testovacieho prostredia
    a. Inštalácia Hadoop, HDFS, MapReduce a ďalších prvkov podľa návrhu testovacieho prostredia
    b. „Smoke test“ testovacieho prostredia použitím vzoriek MapReduce, Pig a Hive
    c. Nasadenie programu na výkon testu

Záverečné zhrnutie

Tento dokument sumarizuje hlavné nálezy vzťahu Big Data a oficiálnej štatistiky a zároveň ponúka odporúčania ďalších krokov smerom k lepšiemu využitiu Big Data v oficiálnej štatistike. Big Data majú rozhodne potenciál produkovať relevantnejšie a včasnejšie štatistiky ako tradičné štatistické zdroje, ktoré sú v súčasnosti takmer exkluzívne postavené na prieskumoch a získavaní administratívnych dát zo štátnej a verejnej správy. V posledných dvoch rokoch prebiehajú v rámci EU intenzívne diskusie o identifikácii možností, ktoré Big Data prinášajú oficiálnej štatistike a zároveň o hlavných strategických a metodických problémoch, ktoré Big Data predstavujú pre oficiálnu štatistiku. Jedným zo záverov týchto debát bolo Scheveningenske Memorandum, ktoré zdôrazňuje potrebu implementácie techník a postup pre spracovanie Big Data v rámci štatistických úradov a definuje základné postupy pre dosiahnutie tohto cieľa či už na úrovni EU alebo regionálnej úrovni.

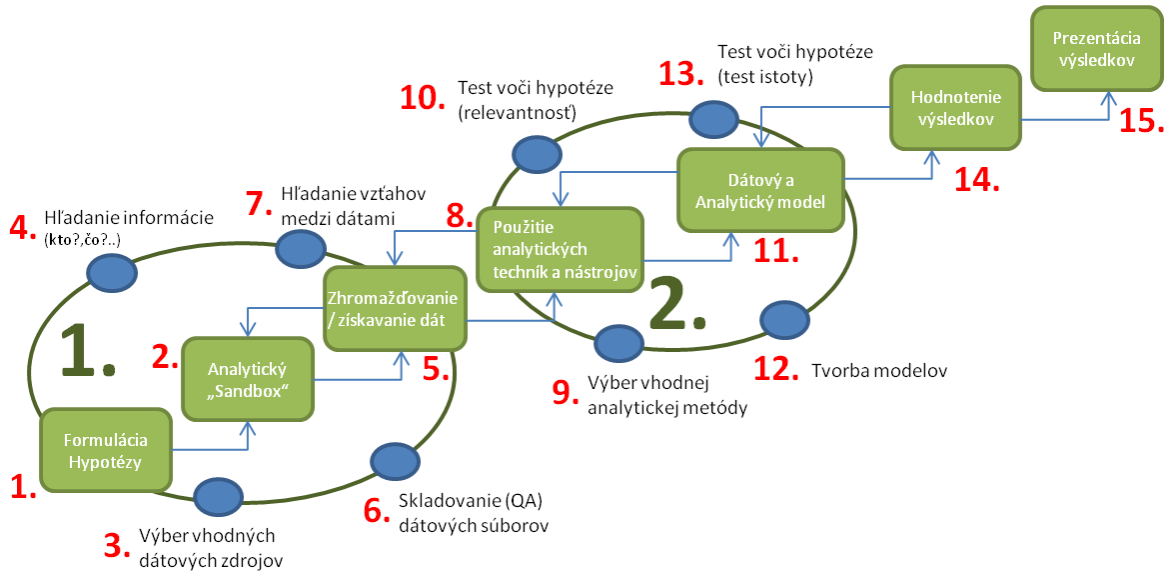
V rámci plnenia úlohy zadanej ŠÚSR sa nám na INFOSTATEe podarilo v priebehu posledného roka zmapovať všetky hlavné iniciatívy štatistických úradov (v rámci EU ako aj zaujímavé mimo-EU projekty), pár zaujímavých projektov je uvedených v tabuľke č.1.

Tabuľka č.1

Štatistický úrad	Názov projektu	Popis
EUROSTAT	Index spotrebiteľských cien tovarov a služieb - zisťovaných na internete.	Vývoj nástroja na monitorovanie spotrebiteľských cien jednotlivých reprezentantov zisťovaných vo vybranej sieti e-shopov a prevádzok služieb poskytovaných na internete. Podobnosť s projektom "The Billion Prices Project" http://bpp.mit.edu/
EUROSTAT	Štatistika cestovnéhop ruchu založená na dátach o polohe	Na základe dát poskytnutých mobilnými operátormi sú generované štatistiky o polohe a cestovnom ruchu.
ISTAT	Prieskum o využívaní informačných a komunikačných technológií v podnikoch	Prepojenie dát získaných z internetu pomocou "Web Scraping" a "Text Mining" s výsledkami oficiálneho prieskumu "Prieskum o využívaní informačných a komunikačných technológií v podnikoch"
ISTAT	Sledovanie krátkodobej migrácie na území povodia vodných tokov	Sledovanie pohybu rezidentov a návštevníkov na území povodia vodných tokov *(administratívne definovaných)* pomocou priestorových súradníc mobilných telefónov.
Štatistický úrad Slovinskej Republiky	Populačné štatistiky s využitím priestorových súradníc mobilných telefónov.	Využitie priestorových súradníc mobilných telefónov na zlepšenie oficiálnej štatistiky najmä pokiaľ ide mobilitu obyvateľstva.
Holandský Štatistický Úrad	Na Holandskom Štatistickom úradu Holandska bolo vykonaných niekoľko prípadových štúdií na "Big Data".	Zdroje dát, ktoré boli sledované na vhodnosť použitia pre oficiálnu štatistiku: 1. Záznamy z elektronického riadiaceho systému dopravy, 2. Dáta z mobilných telefónov, 3. Správy v sociálnych médiách.

Po podrobnej analýze Big Data projektov bola v priebehu roka pri riešení úlohy postupne nadefinovaná metodika pre spracovanie Big Data v podmienkach ŠÚSR (obrázok č.1). Vznikla na základe pozorovania vybraných projektov štatistických úradov a metodík dostupných v komerčnej alebo akademickej sfére v oblasti spracovania Big Data.

Obrázok č.1.



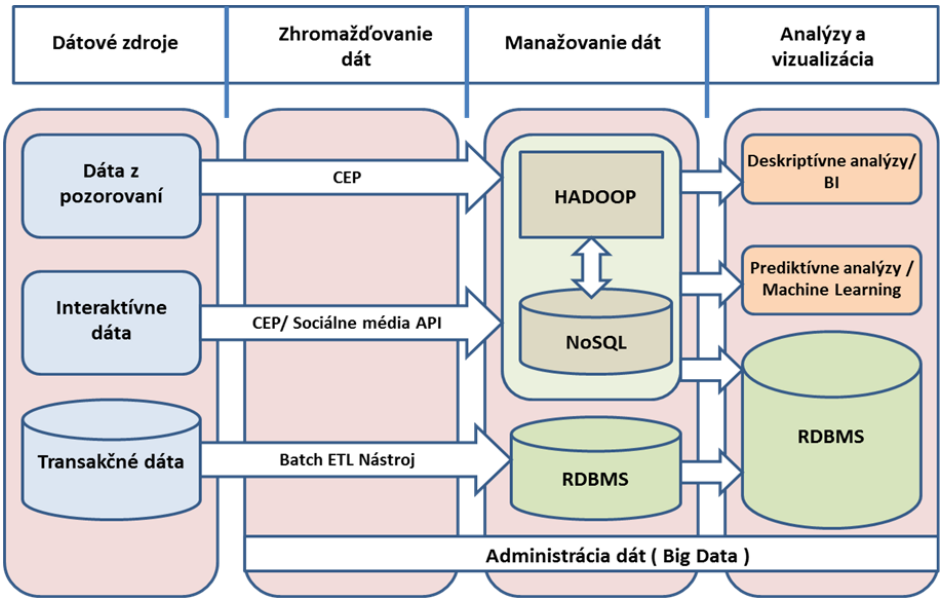
Ďalším krokom bola definícia všeobecnej architektúry Big Data (obrázok č.2). Určenie hlavných postupov, techník a technológií pri spracovaní Big Data, ktoré boli popísané v širokom rozsahu. Obsiahlejšie boli popísané:

HADOOP - softwarový open source framework podporovaný Apache Foundation, ktorý umožňuje distribuované spracovanie veľkých množstiev dát použitím jednoduchých programovacích modelov. Má schopnosť škálovať z jediného až na tisíce počítačov/serverov, z ktorých každý prispieva výpočtovým výkonom a úložným priestorom. Zároveň zabezpečuje vysokú dostupnosť dát, pričom dostupnosť nie je zabezpečená na hardvérovej, ale na softvérovej úrovni Hadoop.

NoSQL - „nie len SQL“ (Not only SQL). NoSQL nedefinuje konkrétnu technológiu alebo riešenie, je všeobecným názvom, ktorý zahŕňa všetky riešenia, ktoré nie sú postavené na báze relačného modelu. Spracované dáta v NoSQL môžu byť štruktúrované, neštruktúrované, alebo semi-štrukturované. Pracuje sa s distribuovanou architektúrou a s dátami, ktoré sú redundantným spôsobom uložené na niekoľkých serveroch. Týmto spôsobom je možné ľahko škálovať systém, pridávaním ďalších serverov, tým vzniká tolerancia k prípadným výpadkom.

Machine Learning - oblasť počítačovej vedy (PV) a umelej inteligencie (AI), ktorá pojednáva o systémoch, ktoré sú schopné učiť sa z dát, ako len explicitne nasledovať programové zadanie. Okrem PV a AI je Machine Learning silno previazaný aj so štatistikou, štúdiom algoritmov a optimalizáciou systémov. Cieľom je poskytnúť čo najpresnejšie predikcie rôznych druhov a pre rôzne účely napr. odhaľovanie podvodov, produktové odporúčania, rôzne druhy segmentácií, rozoznávanie hovoreného slova alebo voľne písaných textov a pod.. Pričom dôraz je kladený na prediktívne analýzy v reálnom čase a vysokú prispôsobivosť systému.

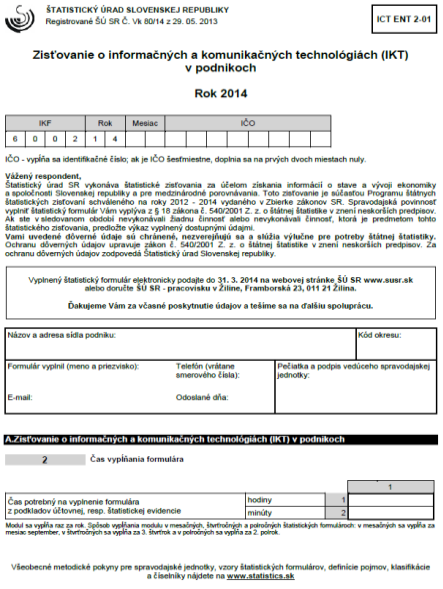
Obrázok č.2



V priebehu roka bolo vykonaných aj niekoľko zberov a analýz dát z internetu pre lepšiu ilustráciu využitia Big Data. Testovaných bolo napríklad viacero internetových zdrojov (blogy, komentáre a iné dáta generované človekom) a pre malú prípadovú štúdiu bol vybraný Twitter najme kvôli užívateľsky veľmi prijateľnému API. Množstvu neštruktúrovaných dát ,ktoré okrem písmen obsahujú aj čísla a #Hashtag, ktorý sa používa na opis daného tweetu. Zber dát a analýza dát prebiehala v reálnom čase. Pri analýze dát sa jednalo o analýzu neštruktúrovaných dát – textu, kde sa sledoval výskyt jednotlivých slov ako aj korelácie medzi nimi.

Pre analýzu praktického využitia Big Data v oficiálnej štatistike bol vybraný formulár „Zisťovanie o informačných a komunikačných technológiách (IKT) v podnikoch“.

Štatistický úrad SR vykonáva štatistické zisťovania za účelom získania informácií o stave a vývoji ekonomiky a spoločnosti Slovenskej republiky a pre medzinárodné porovnávania. Toto zisťovanie je súčasťou Programu štátnych štatistických zisťovaní schváleného na roky 2012 - 2014 vydaného v Zbierke zákonov SR. Spravodajská povinnosť vyplniť štatistický formulár vyplýva z § 18 zákona č. 540/2001 Z. z. o štátnej štatistike v znení neskorších predpisov. Po podrobnej analýze formulára a preskúmaní dostupnosti dopytovaných informácií na internete boli vybrané tematické okruhy na základe otázok 13 až 24 - Zisťovanie o informačných a komunikačných technológiách (IKT) v podnikoch.


ŠTATISTICKÝ ÚRAD SLOVENSKEJ REPUBLIKY
Registrované ŠÚ SR Č. Vk 80/14 z 29. 05. 2013

ICT ENT 2-01

Zisťovanie o informačných a komunikačných technológiách (IKT) v podnikoch

Rok 2014

IKF			Rok		Mesiac		IČO								
6	0	0	2	1	4										

IČO - vypĺňa sa identifikačné číslo; ak je IČO šesťmiestne, doplnia sa na prvých dvoch miestach nuly.

Vážený respondent,
Štatistický úrad SR vykonáva štatistické zisťovania za účelom získania informácií o stave a vývoji ekonomiky a spoločnosti Slovenskej republiky a pre medzinárodné porovnávania. Toto zisťovanie je súčasťou Programu štátnych štatistických zisťovaní schváleného na roky 2012 - 2014 vydaného v Zbierke zákonov SR. Spravodajská povinnosť vyplniť štatistický formulár Vám vyplýva z § 18 zákona č. 540/2001 Z. z. o štátnej štatistike v znení neskorších predpisov. Ak ste v sledovanom období nevykonávali žiadnu činnosť alebo nevykonávali žiadnu činnosť, ktorá je predmetom tohto štatistického zisťovania, predložte výkaz vyplnený dostupnými údajmi.
Vami uvedené dôverné údaje sú chránené, nezverejňujú sa a slúžia výlučne pre potreby štátnej štatistiky. Ochranu dôverných údajov upravuje zákon č. 540/2001 Z. z. o štátnej štatistike v znení neskorších predpisov. Za ochranu dôverných údajov zodpovedá Štatistický úrad Slovenskej republiky.

Vyplnený štatistický formulár elektronicky podajte do 31. 3. 2014 na webovej stránke ŠÚ SR www.susr.sk alebo doručte ŠÚ SR - pracovisku v Žiline, Framborská 23, 011 21 Žilina.

Ďakujeme Vám za včasné poskytnutie údajov a tešíme sa na ďalšiu spoluprácu.

Názov a adresa sídla podniku:		Kód okresu:
Formulár vyplnil (meno a priezvisko):	Telefón (vrátane smerového čísla):	Pečiatka a podpis vedúceho spravodajskej jednotky:
E-mail:	Odoslané dňa:	

A.Zisťovanie o informačných a komunikačných technológiách (IKT) v podnikoch

2 Čas vypĺňania formulára

		1
Čas potrebný na vyplnenie formulára z podkladov účtovnej, resp. štatistickej evidencie	hodiny 1	
	minúty 2	

Modul sa vypĺňa raz za rok. Spôsob vypĺňania modulu v mesačných, štvrťročných a polročných štatistických formulároch: v mesačných sa vypĺňa za mesiac september, v štvrťročných sa vypĺňa za 3. štvrťrok a v polročných sa vypĺňa za 2. polrok.

Všeobecné metodické pokyny pre spravodajské jednotky, vzory štatistických formulárov, definície pojmov, klasifikácie a číselníky nájdete na www.statistics.sk

Tematické okruhy nadefinovali hlavne okruhy otázok pre automatický zber internetu:

- Webstránka obsahuje eshop?
- Webstránka obsahuje linky na sociálne siete ako facebook alebo twitter?
- Webstránka obsahuje certifikát bezpečnosti (SSL) ?
- Webstránka obsahuje multimédia ako video alebo hudbu?
- Na webstránke sa nachádzajú kontaktne údaje - email a telefón spoločnosti?

Automatický zber z internetu bol realizovaný pomocou techniky nazývanej WebScraping - proces automatického zberu dát z internetu. Samotný webscrapingový nástroj bol naprogramovaný v jazyku PERL. Na základe dodaného excelovského súboru, ktorý obsahoval 4247 celkovo štatistických jednotiek nástroj postupne načítaval jednotlivé jednotky a vyhľadával ich na internete s cieľom zodpovedať nadefinované okruhy otázok. Konečná štatistika nebola veľmi presvedčivá v prospech automatizácie zberov údajov z internetu. Celkovo nástroj identifikoval 1394 jednotiek na internete pričom ani v jednom prípade neboli zodpovedané všetky otázky pre definované tematické okruhy. Uvedený výsledok vypovedá viac o stále pomerne nízkej prezentácií firiem na internete ako o samotnej efektivite nástroja. Samotný nástroj dosahoval pomerne vysokú presnosť a efektivitu zberu a analýzy dát.

Na základe poznatkov získaných počas riešenia úlohy je možné konštatovať ,že v súčasnosti je vhodné vnímať Big Data skôr ako podporné dáta pre oficiálne štatistiky. Big Data neposkytujú plnú náhradu za oficiálne zbery či už z pohľadu rozsahu alebo kvality zozbieraných údajov. Môžu ,ale pri správnom nastavení znižovať náklady na vybrané štatistické zisťovania, znižovať zaťaženie sledovaných štatistických jednotiek (firiem) pri poskytovaní sledovaných údajov a poskytnúť väčšiu flexibilitu pri následných analýzach.
Porovnanie hlavných rozdielov pre tvorbu oficiálnych a štatistík a spracovaní Big Data je uvedené v tabuľke:

Oficiálna štatistika	Big Data
1. Štruktúrované dáta	1.Najmä neštruktúrované dáta
2. Jasný koncept a metodológia	2. Väčsinou bez vopred nadefinovanej metodológie a konceptu
3. Regulované	3. Neregulované
4. Macro-úroveň ,ale typicky založená na veľkých objemoch primárnych dát	4. Micro-úroveň založená na enormných objemoch dát rôznej frekvencie, druhu a rýchlosti
5. Vysoké náklady	5. Vo všeobecnosti nízke alebo žiadne náklady
6. Centralizvané; v definovanom časovom úseku	6. Rozptýlené; v realnom čase

Výhodou Big Data do blízkej budúcnosti môže byť definícia úplne nových druhov štatistických zisťovaní napríklad pri použití dát zo sociálnych sietí alebo zo sledovania dopravy, mýtneho systému.
Veľkou nevýhodou sú stále pomerne vysoké náklady na technickú implementáciu Big Data riešení ako aj pomerne vysoké kvalifikačné nároky na ľudské zdroje

Použité zdroje

Knihy:

Big Data Imperatives: Enterprise ‘Big Data’ Warehouse, ‘BI’ Implementations and Analytics (The Expert's Voice) link
Learning R link
Machine Learning for Hackers link
Machine Learning in Action link
Big Data for Dummies link
Big Data Glossary link
The Definitive Guide to MongoDB: The NoSQL Database for Cloud and Desktop Computing (Expert's Voice in Open Source) link
Getting started with NoSQL link
Hadoop Operations link
MapReduce Design Patterns link
Hadoop Definitive Guide link
Mining the Social Web link
RESTful Web APIs link
Data Science for Business link
Cloud Architecture Patterns link
McKinsey&Company – Big data: The next frontier for innovation, competition and productivity link

Online kurzy:

MIT – Analytics Edge link
MIT – Big Data and Social Physics link
Stanford – Machine Learning link
University of Washington – Introduction to Data Science link
Johns Hopkins University – Getting and Cleaning Data link
Johns Hopkins University – Statistical Inference link
Johns Hopkins University – Practical Machine Learning link
Johns Hopkins University – Exploratory Data Analysis link
Johns Hopkins University – R programming link
Johns Hopkins University – The Data Scientist’s Toolbox link

Internetové zdroje:

Hadoop
http://hadoop.apache.org/
http://www.cloudera.com/content/cloudera/en/about/hadoop-and-big-data.html
https://www.mapr.com/products/apache-hadoop
http://hortonworks.com/hadoop/

https://github.com/apache/hadoop-common
http://blog.cloudera.com/blog/2014/05/how-apache-hadoop-yarn-ha-works/
http://hortonworks.com/hadoop/yarn/
http://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html
https://developer.yahoo.com/hadoop/tutorial/module4.html
http://www.cs.colorado.edu/~kena/classes/5448/s11/presentations/hadoop.pdf
http://www.cloudera.com/content/cloudera/en/products-and-services/cdh/hdfs-and-mapreduce.html
https://hadoop.apache.org/docs/r0.18.0/hdfs_design.pdf
http://www.aosabook.org/en/hdfs.html
http://hortonworks.com/hadoop/hdfs/
https://developer.yahoo.com/hadoop/tutorial/module2.html
http://static.googleusercontent.com/media/research.google.com/es/us/archive/mapreduce-osdi04.pdf
http://cecs.wright.edu/~tkprasad/courses/cs707/ProgrammingHadoop.pdf
http://mapreduce.sandia.gov/index.html
https://github.com/apache/hadoop-mapreduce

NoSQL

http://nosql-database.org/
http://www.mongodb.com/nosql-explained
http://www.aosabook.org/en/nosql.html
http://www.techrepublic.com/blog/10-things/10-things-you-should-know-about-nosql-databases/
http://highlyscalable.wordpress.com/2012/03/01/nosql-data-modeling-techniques/
http://www.oracle.com/us/products/database/nosql/overview/index.html
http://www.christof-strauch.de/nosqldbs.pdf
http://www.networkworld.com/article/2160905/tech-primers/a-vendor-independent-comparison-of-nosql-databases--cassandra--hbase--mongodb--riak.html

1. Key-Value Stores:
    1. http://www.devshed.com/c/a/MySQL/Database-Design-Using-KeyValue-Tables/
    2. http://antirez.com/post/Sorting-in-key-value-data-model.html
    3. http://stackoverflow.com/questions/3554169/difference-between-document-based-and-key-value-based-databases
    4. http://dbmsmusings.blogspot.com/2010/03/distinguishing-two-major-types-of_29.html
2. BigTable-style Databases:
    1. http://www.slideshare.net/ebenhewitt/cassandra-datamodel-4985524
    2. http://www.slideshare.net/mattdennis/cassandra-data-modeling
    3. http://nosql.mypopescu.com/post/17419074362/cassandra-data-modeling-examples-with-matthew-f-dennis
    4. http://s-expressions.com/2009/03/08/hbase-on-designing-schemas-for-column-oriented-data-stores/
    5. http://jimbojw.com/wiki/index.php?title=Understanding_Hbase_and_BigTable
3. Document Databases:
    1. http://www.slideshare.net/mongodb/mongodb-schema-design-richard-kreuters-mongo-berlin-preso

2. http://www.michaelhamrah.com/blog/2011/08/data-modeling-at-scale-mongodb-mongoid-callbacks-and-denormalizing-data-for-efficiency/
3. http://seancribbs.com/tech/2009/09/28/modeling-a-tree-in-a-document-database/
4. http://www.mongodb.org/display/DOCS/Schema+Design
5. http://www.mongodb.org/display/DOCS/Trees+in+MongoDB
6. http://blog.fiesta.cc/post/11319522700/walkthrough-mongodb-data-modeling

4. Full Text Search Engines:
    1. http://www.searchworkings.org/blog/-/blogs/query-time-joining-in-lucene
    2. http://www.lucidimagination.com/devzone/technical-articles/solr-and-rdbms-basics-designing-your-application-best-both
    3. http://blog.griddynamics.com/2011/07/solr-experience-search-parent-child.html
    4. http://www.lucidimagination.com/blog/2009/07/18/the-spanquery/
    5. http://blog.mgm-tp.com/2011/03/non-standard-ways-of-using-lucene/
    6. http://www.slideshare.net/MarkHarwood/proposal-for-nested-document-support-in-lucene
    7. http://mysolr.com/tips/denormalized-data-structure/
    8. http://sujitpal.blogspot.com/2010/10/denormalizing-maps-with-lucene-payloads.html
    9. http://java.dzone.com/articles/hibernate-search-mapping-entit
5. Graph Databases:
    1. http://docs.neo4j.org/chunked/stable/tutorial-comparing-models.html
    2. http://blog.neo4j.org/2010/03/modeling-categories-in-graph-database.html
    3. http://skillsmatter.com/podcast/nosql/graph-modelling
    4. http://www.umiacs.umd.edu/~jimmylin/publications/Lin_Schatz_MLG2010.pdf
6. Demensionality Reduction:
    1. http://www.slideshare.net/mmalone/scaling-gis-data-in-nonrelational-data-stores
    2. http://blog.notdot.net/2009/11/Damn-Cool-Algorithms-Spatial-indexing-with-Quadtrees-and-Hilbert-Curves
    3. http://www.trisis.co.uk/blog/?p=1287

Machine Learning

http://cran.r-project.org/web/views/MachineLearning.html
http://archive.ics.uci.edu/ml/
http://www.cs.waikato.ac.nz/ml/
http://hunch.net/
http://mlg.eng.cam.ac.uk/
http://learning.cs.toronto.edu/
http://aitopics.org/topic/machine-learning
http://eval.symantec.com/mktginfo/enterprise/white_papers/b-dlp_machine_learning.WP_en-us.pdf
http://www.cs.cmu.edu/~tom/pubs/MachineLearning.pdf
http://www.kaggle.com/

PRÍLOHY

Príloha č.1

Version 1.05_27.1.2014

Obsah

Úvod

Big Data nie je technológia skôr fenomén posledných rokov, ktorý vznikol z obrovského množstva neštruktúrovaných dát ,ktoré denne vzniknú a je ich problém spracovať tradičnými metódami spracovania dát. Tento dokument sa zaoberá vzťahom Big Data k Oficiálnej štatistike preto nebude vysvetľovať základné definície, vzťahy a procesy *(jednoduchá definícia Big Data MIT Technology review (predklad webovej stránky))*.

Big Data majú do budúcna potenciál produkovať relevantnejšie a včasnejšie štatistiky ako tradičné štatistické zdroje, ktoré sú v súčasnosti takmer exkluzívne postavené na prieskumoch a získavaní administratívnych dát zo štátnej a verejnej správy.

Tento dokument je inicializačným dokumentom k projektu - Preskúmať potenciál „Big data“ dátových zdrojov ako novú príležitosť a výzvu pre oficiálnu štatistiku. Sleduje vývoj vo svete v oblasti big data a ich vzťahu k tradičnej oficiálnej štatistike. Ponúka príklady zo sveta a prvotný návrh „univerzálnej“ metodiky spracovania Big Data.

Pojmy použité v dokumente vo vzťahu k „Big Data“

Web Scraping - (počítačový softvér) technika získavania informácií z webových stránok. Zameriava na transformáciu neštruktúrovaných dát na webe, zvyčajne vo formáte HTML, do štruktúrovaných dát, ktoré môžu byť uložené a analyzované v lokálnej databáze alebo tabuľkovom procesore.

Text Mining - získavania informácií z textu na základe zadefinovaných parametrov. Text Mining zvyčajne zahŕňa proces štruktúrovania vstupného textu, definíciu väzieb a významu v rámci štruktúrovaných dát, hodnotenie a interpretácia výstupu. Medzi typické úlohy Test Mining patrí textová kategorizácia, zlučovanie textu, extrakcia konceptu entity/sledovaného textu, tvorba taxonómie, analýza sentimentu, sumarizácia dokumentov a modelovanie vzťahov medzi sledovanými entitami.

Machine learning – odbor AI (umelej inteligencie), ktorý sa zaoberá tvorbou a štúdium systémov, ktoré sa vedia učiť z dát. Príklad: rozlišovanie e-mailových správ medzi spamom a non-spamom.

Cloud – Cloud computing je pojem používaný na popis prepojenia počítačov pripojených pomocou real-time komunikačnej siete, ako je napr. Internet. Znamená schopnosť spustiť program alebo aplikáciu na mnohých pripojených počítačov súčasne a z rôznych lokalít.
Základné formy:

- IaaS – Infraštruktúra ako služba
- PaaS – Platforma ako služba
- SaaS – Software ako služba

NoSQL – databáza NoSQL poskytuje mechanizmus pre ukladanie a vyhľadávanie dát a je modelovaná v iných ako tabuľkových vzťahoch používaných v relačných databázach. Zvyčajne ma jednoduchú konštrukciu, horizontálne škálovanie a lepšiu kontrolovateľnosť ako RDBMS.

Hadoop – Apache Hadoop je open-source software framework na skladovanie a spracovanie veľkých objemoch dát pri nízkych nákladoch oproti klasickým riešeniam.
Apache Hadoop sa skladá z nasledujúcich modulov:

- Hadoop Common - obsahuje knižnice a nástroje
- Hadoop Distributed File System (HDFS) - distribuovaný súborový systém, poskytuje vysokú priepustnosť prístupu k aplikačným dátam
- Hadoop YARN - platforma zodpovedná za riadenie úloh a zdrojov.
- Hadoop MapReduce - systém pre paralelné spracovanie veľkých dátových súborov

MapReduce – je programovací model pre spracovanie veľkých dátových súborov paralelným, distribuovaným algoritmom. Skladá sa:

- Map () procesu - vykonáva filtrovanie a triedenie
- Reduce () procesu - vykonáva súhrnnú funkciu

API - Application Programming Interface (API) - Definuje vzájomnú komunikáciu jednotlivých software-ových komponentov. Okrem prístupu k DB a počítačovému HW sa používa aj pre prácu v GUI (grafickom rozhraní).

Štatistika a Big Data vo svete

New York Times napísal vo februári 2012 *„Na základe pozorovania sa dá konštatovať, že sledovanie vybraných slov a ich intenzity používania vo vyhľadávači google má vyššiu schopnosť prognózovania stavebnej produkcie v nasledujúcom štvrťroku ako výstupy oficiálnej štatistiky".*

Uvedené vyhlásenie presne charakterizuje stav, v ktorom sa oficiálna štatistika ocitla vo vzťahu k stále novým zdrojom dát a ich spracovaniu. Až na pár iniciatív štatistické úrady zmeškali nástup Big Data či už v oblasti ich získavania alebo analýzy.

V USA je zaujímavé sledovať pomerne kritický postoj odbornej verejnosti voči slabej angažovanosti oficiálnych spracovateľov štatistických dát (napr. Census, BLS) na využívaní Big Data. Čo sa môže zdať zvyšku sveta pomerne prekvapivé, keďže administratíva Baracka Obamu si dala spracovanie Big Data už v roku 2012 medzi svoje priority, čo následne zdôraznila aj v Novembri 2013 spustením ďalšej iniciatívy v tejto oblasti. Implementácia Big Data prebieha horizontálne celou štátnou a verejnou správou a jednotlivé úrady a agentúry sa snažia implementovať nové metódy a technológie do svojej praxe. Lídrom v spracovaní Big Data v USA tomto smere je NSA (Národný Bezpečnostný Úrad) a to nielen kvôli už zverejnením kauzám.

Nekorunovanými lídrami v Ázii v spracovaní Big Data sú Južná Kórea a Japonsko, ktorí už majú svoje prvé projekty za sebou a v súčasnosti prebiehajú ďalšie (*stále ale na úrovni testovania a „proof of concept"*). Čínsky štatistický úrad v spolupráci s konzorciom technologických firiem ohlásil koncom roka 2013 svoju prvú Big Data iniciatívu zameranú na sledovanie inflácie.

Austrália a Nový Zéland patria k priekopníkom v spracovaní Big Data a projekty v uvedených krajinách majú už aj rozsiahle praktické využitie. Rozsah niektorých projektov či už z pohľadu dopadu daného projektu na sledovanú problematiku alebo veľkosti je porovnateľný s najúspešnejšími projektmi v komerčnej sfére v danom regióne.

V rámci EU sa v priebehu minulého roka viedli diskusie na globálnej úrovni o identifikácii možností, ktoré Big Data prinášajú oficiálnej štatistike a zároveň o hlavných strategických a metodických problémoch, ktoré Big Data predstavujú pre oficiálnu štatistiku. Záverom týchto debát bolo Scheveningenske Memorandum :

Eurostat – (CORS - Collaboration in Research and Methodology for Official Statistics)
Scheveningen memorandum plné znenie memoranda

V jednotlivých členských štátoch prebiehali a prebiehajú aj individuálne projekty. Za zmienku stoji predovšetkým Holandsky štatistický úrad, ktorý už ma za sebou 3 projekty so spracovaním Big Data.

Nasledujúca tabuľka ponúka prehľad Big Data iniciatív prebiehajúcich v súčasnosti v rámci štatistických úradov. Spomenuté sú iba iniciatívy ku ktorým bolo možné dohľadať relevantné informačné zdroje. Iniciatívy v oblasti Big Data minimálne vo forme verejnej diskusie prebiehajú aj na iných štatistických úradoch ako napr. Indonézia, Jamajka alebo Filipíny.

Štatistický úrad	Názov projektu	Popis	"Big Data"				
			Rozsah projektu	Typ "Big Data"	Zdroj - sektor	Prístup k dátam	Nástroje a metódy na analýzu "Big Data"
Australsky Štatistický úrad	Spracovanie dát zo satelitného snímania za účelom odhadu využitia poľnohospodárskej pôdy	Sledovanie využitia poľnohospodárskej pôdy za účelom odhadu podielu obhospodarovanej pôdy a výnosu z plodín.	Národný	Satelitné snímky	Verejný sektor	Prvotné dáta (štruktúrované a neštruktúrované) sú spracovávané mimo štatistického úradu (napr. v rámci systémov dodávateľa, alebo v "cloude" - spoločný výskum s univerzitami).	"Machine learning" je použitý na extrakciu a klasifikáciu. Nadefinované dátové štruktúry umožňujú správu a analýzu časovo-priestorových geoúdajov.
Australsky Štatistický úrad	Spracovanie dát zo sociálnych sieti (sémantické data) - spracovanie dát pre rôzne štatistické využitie	Prepojenie získaných dát s výsledkami oficiálnej štatistiky napr. sčítanie a iné sociálne a demografické štatistiky	Národný	Semantické dáta	Verejný sektor	Pomocou integračných technológií (dátové prepojenia) sú neštruktúrované a polo-štruktúrované dáta prenesené do štatistického úradu na spracovanie.	Pomocou použitia webových sémantických technológií sa vytvorila modelová dátovej sada (OWL / RDF) pre účely analýzu. Na ktoré boli následne techniky "machine learning" (napr. SVM).
EUROSTAT	Index spotrebiteľských cien tovarov a služieb - zisťovaných na internete.	Vývoj nástroja na monitorovanie spotrebiteľských cien jednotlivých reprezentantov zisťovaných vo vybranej sieti e-shopov a prevádzok služieb poskytovaných na internete. Podobnosť s projektom "The Billion Prices Project" http://bpp.mit.edu/	Medziná-rodný	Rôzne neštruktúro-vané dáta (najčastejšie - HTML format)	Súkromný sektor	Dáta získané priamo z internetu použitím „web scraped“ open-source nástroja z verejne dostupných zdrojov.	Zatiaľ nezadefinované
EUROSTAT	Štatistika cestovnéhop ruchu založená na dátach o polohe	Na základe dát poskytnutých mobilnými operátormi sú generované štatistiky o polohe a cestovnom ruchu.	Medziná-rodný	Dáta poskytnuté mobilnými operátormi	Súkromný sektor	Pomocou integračných technológií (dátové prepojenia) sú neštruktúrované a polo-štruktúrované dáta prenesené do štatistického úradu na spracovanie.	Zatiaľ nezadefinované
ISTAT	Prieskum o využívaní informačných a komunikačných technológií v podnikoch	Prepojenie dát získaných z internetu pomocou "Web Scraping" a "Text Mining" s výsledkami oficiálneho prieskumu "Prieskum o využívaní informačných a komunikačných technológií v podnikoch"	Národný	Rôzne neštruktúro-vané dáta (najčastejšie - HTML format)	Súkromný sektor	Pomocou integračných technológií (dátové prepojenia) sú neštruktúrované a polo-štruktúrované dáta prenesené do štatistického úradu na spracovanie.	Pre "Web Scraping" aj "Text Mining" sú testované rôzne analytické nástroje s cieľom vybrať najvhodnejší nástroj.

Štatistický úrad	Názov projektu	Popis	"Big Data"				
			Rozsah projektu	Typ "Big Data"	Zdroj - sektor	Prístup k dátam	Nástroje a metódy na analýzu "Big Data"
ISTAT	Sledovanie krátkodobej migrácie na území povodia vodných tokov	Sledovanie pohybu rezidentov a návštevníkov na území povodia vodných tokov *(administratívne definovaných)* , pomocou priestorových súradníc mobilných telefónov.	Národný	Dáta poskytnuté mobilnými operátormi	Verejný sektor/ Súkromný sektor	Prvotné dáta (štruktúrované a neštruktúrované) sú spracovávané mimo štatistického úradu (v rámci systémov dodávateľa a v "cloude" - spoločný výskum s University of Pisa s cieľom spracovnia dát z mobilných telefónov).	"Machine learning" - (SOM - Self-Organising Maps)
Štatistický úrad NZ	Sledovanie krátkodobej migrácie počas a po živelnej pohrome. *(Zemetrasenie Christchurch, Nový Zéland 22.feb.2011)*	Štatistický úrad NZ vykonal analýzu v spolupráci s telekomunikačnými spoločnosťami, ministerstvom civilnej obrany a krízového riadenia, a ďalších agentúr. Sledovaná bola intenzita hovorov a priestorové súradnice anonymizovaných Christchurch mobilných telefónov 10 dni pred a po zemetrasení v Christchurch 22.februára 2011.	Národný	Dáta poskytnuté mobilnými operátormi	Verejný sektor/ Súkromný sektor	Pomocou integračných technológií (dátové prepojenia) boli neštruktúrované a polo-štruktúrované dáta prenesené do štatistického úradu na spracovanie.	Štandardné štatistické postupy
Štatistický úrad Slovinskej Republiky	Populačné štatistiky s využitím priestorových súradníc mobilných telefónov.	Využitie priestorových súradníc mobilných telefónov na zlepšenie oficiálnej štatistiky najmä pokiaľ ide mobilitu obyvateľstva.	Národný	Dáta poskytnuté mobilnými operátormi	Súkromný sektor	Pomocou integračných technológií (dátové prepojenia) sú neštruktúrované a polo-štruktúrované dáta prenesené do štatistického úradu na spracovanie.	Zatiaľ nezadefinované
Holandský Štatistický Úrad	Na Holandskom Štatistickom úradu Holandska bolo vykonaných niekoľko prípadových štúdií na "Big Data".	Zdroje dát, ktoré boli sledované na vhodnosť použitia pre oficiálnu štatistiku: 1. Záznamy z elektronického riadiaceho systému dopravy, 2. Dáta z mobilných telefónov, 3. Správy v sociálnych médiách.	Národný	Dáta poskytnuté mobilnými operátormi/ Rôzne neštruktúro-vané dáta	Verejný sektor/ Súkromný sektor	Pomocou integračných technológií (dátové prepojenia) boli neštruktúrované a polo-štruktúrované dáta prenesené do štatistického úradu na spracovanie.	Využitie programovacích jazykov Python a R

Štatistický úrad	Názov projektu	Popis	"Big Data"				
			Rozsah projektu	Typ "Big Data"	Zdroj - sektor	Prístup k dátam	Nástroje a metódy na analýzu "Big Data"
Štatistický Úrad Južná Korea (KOSTAT)	Na KOSTAT-e boli vykonané 2 prípadových štúdií na "Big Data": 1. Definovanie rizikových faktorov pre mládež na základe dát zo sociálnych sietí 2. Analýza dát z médií ako podpora pre Index priemyselnej produkcie	1. Definovanie rizikových faktorov pre mládež na základe dát zo sociálnych sietí *a. Riziko samovraždy – sledovanie rizikových faktorov a správania s cieľom lepšie pochopiť okolnosti a psychologické stavy mládeže* *b. Šírenie škodlivého obsahu – cez populárne URL* 2. Analýza dát z médií ako podpora pre Index priemyselnej produkcie zameraná na 4 druhy priemyslu: *C21 (Pharmaceuticals)* *C24 (Primary Metals)* *C26 (Electronic Components, …)* *C28 (Electric Equipment)*	Národný	Rôzne neštruktúro-vané dáta (PDF,DOC a pod.). Ďalej JSON a XML.	Verejný sektor/ Súkromný sektor	Dáta získané priamo z internetu z verejne dostupných zdrojov použitím "web scraped“ a "web-crawling" open-source nástrojov.	Využitie programovacích jazykov Python a R
USA	Marec 2012 ***Big data Iniciatíva*** Obamovej Adminitratívy November 2013 Nová iniciatíva ***Big Data to Knowledge***	Beží paralelne niekoľko projektov v rámci štátnej a verejnej správy. V marci 2012 Obama administratíva oznámila Big Data iniciatívu (link), pričom v novembri 2013 bola táto iniciatíva ďalej rozšírená smerom k znalostnej ekonomike (kompletný list projektov). V oblasti štatistiky beží niekoľko projektov so zameraním na: Štatistiky v stavebníctve, Štatistiky maloobchodu a služieb a Sčítanie 2020.	Národný	Rôzne zdroje	Verejný sektor/ Súkromný sektor	Použité sú rôzne techniky a technológie	Použité sú rôzne techniky a technológie

Typové rozdelenie Big Data z pohľadu Oficiálnej Štatistiky

Členenie podľa UNECE Divízia Štatistiky - Projektový team „Big Data“, Jún 2013

1. Dáta generované človekom: väčšinou neštruktúrované dáta.
    - 1100. Sociálne siete: Facebook, Twitter, Tumblr atď.
    - 1200. Blog a komentáre
    - 1300. Osobné dokumenty
    - 1400. Obrázky: Instagram, Flickr, Picasa atď.
    - 1500. Videá: Youtube atď.
    - 1600. Vyhľadávania na internete
    - 1700. Mobilné dáta: textové správy
    - 1800. Užívateľmi generované mapy
    - 1900. E-Mail

2. Tradičné obchodné systémy: väčšinou vysoko štruktúrované, zahŕňajúce transakcie, referenčné tabuľky a vzťahy, rovnako ako aj metadáta.
    - 21. Dáta produkované verejnou správou
        - 2110. Zdravotné záznamy
    - 22. Dáta produkované podnikateľským prostredím
        - 2210. Komerčné transakcie
        - 2220. Bankové záznamy/ Finančné trhy
        - 2230. E-commerce
        - 2240. Kreditné karty

3.Strojom generované dáta (Internet of Things) : Zvyčajne štruktúrované a pološtruktúrovane, časť uložená RDBMS.
    - 31. Dáta zo senzorov
        - 311. Fixné/pevné senzory
            - 3111. V domácnostiach
            - 3112. Na sledovanie počasia a znečistenia
            - 3113. Na sledovanie dopravy (vrátane foto-video techniky)
            - 3114. Vo vede a výskume
            - 3115. Bezpečnostné (vrátane foto-video techniky)
        - 312. Mobilné senzory (monitorovacie)
            - 3121. Mobilné zariadenia (priestorové súradnice)
            - 3122. Dopravné prostriedky
            - 3123. Satelitné snímky
    - 32. Dáta z počítačových systémov
        - 3210. Logs
        - 3220. Web logs

Štatistika a Big Data vo svete – postrehy

Je možné postrehnúť, že úspešné iniciatívy v rámci ŠÚ v Big Data (napr. Nový Zeland, Austrália) sa nezačínali diskusiou o technológii ,ale skôr snahou adresovať problémy, riešiť požiadavky, ktoré sa nedali riešiť tradičnými prístupmi. Úspešné projekty majú v počiatočnej fáze zhodné črty najmä v tom, že majú tendenciu začať konkrétnou úzko vymedzenou hypotézou, ktorá sa opiera o jednu z Big Data charakteristík – objem, rýchlosť alebo rôznorodosť. Z toho vyplýva, že nie je vhodné začať systémom – postavíme univerzálnu technickú platformu a dáta potom prídu. Tu je, ale nutne zdôrazniť, že sa jedná o štandardný štatistický prístup (zhora nadol – od hypotézy k modelu) pričom v iných odvetviach je úspešne používaný aj prístup zdola nahor (od dát k hypotéze).

Priemerná dĺžka projektu 12 -13 mesiacov (pričom sa jedná primárne o pilotné projekty), v súkromnej sfére je priemerná dĺžka Big Data projektu 18 mesiacov (podľa IDG Research).

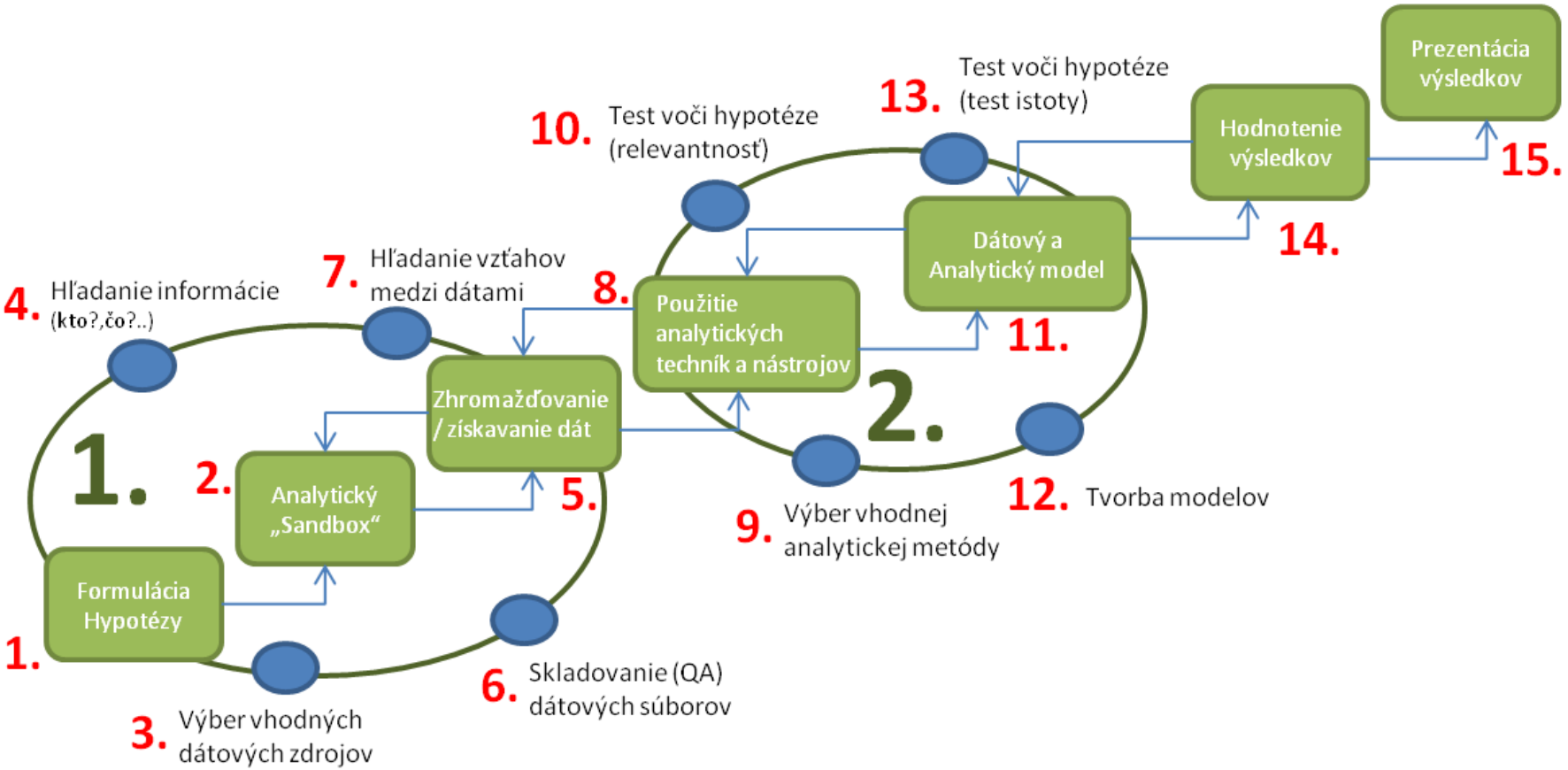
Z pohľadu technologického zabezpečenia je pri spracovaní veľkých objemov dát a aj vďaka svojej flexibilite dominantný Hadoop. Preferovanými formátmi su XML a JSON. Databázy sa používajú rôzne NoSQL DB ako aj tradičné RDMS (aj ako hybrid s NoSQL). Na samotnú analýzu je jednoznačná preferencia programovacích jazykov Python a R (aj pre „riadený/kontrolovaný“ machine learning).

Hlavné bariéry pri implementácii Big Data v rámci Štatistických úradov

1. Legislatíva – ochrana osobných údajov, autorské práva
2. Ochrana súkromia – zabezpečenie „komfortnej“ súkromnej užívateľskej zóny tak, aby vznikla dôvera medzi poskytovateľmi dát a ich spracovateľmi
3. Financie – finančná náročnosť projektov v prostredí štatistických úradov, definícia ROI
4. Manažment – definícia nových smerníc a nariadení v oblasti spracovania a ochrany dát, MDM
5. Ľudské zdroje – systém vzdelávania a rozvoja ľudských zdrojov v oblasti Big Data
6. Metodológia – kvalita dát, rozvoj analytických metód a techník
7. Technologické zabezpečenie – rozvoj Big Data infraštruktúry, IT

V tomto materiály sa bude najväčší priestor venovať definícii metodológie a technologického zabezpečenia.

Návrh metodiky spracovania Big Data v podmienkach ŠÚSR *(v1.07)*



Navrhovaná metodika je prvotným návrhom koncipovaným na základe pozorovania vybraných projektov štatistických úradov a metodík dostupných v komerčnej alebo akademickej sfére v oblasti spracovania Big Data.

Slučka č.1.

Formuluje hypotézu k určitému problému alebo javu, prípadne sa snaží potvrdiť alebo vyvrátiť klasické štatistické zisťovanie, analýzy sociálno-ekonomického a ekologického vývoja. Na základe danej hypotézy sa vytvára analytický sandbox a zhromažďujú sa požadované dáta.

Primárna funkcia slučky č.1 je zhromažďovanie dát a práca s nimi na základe sledovanej hypotézy pričom sa sleduje:

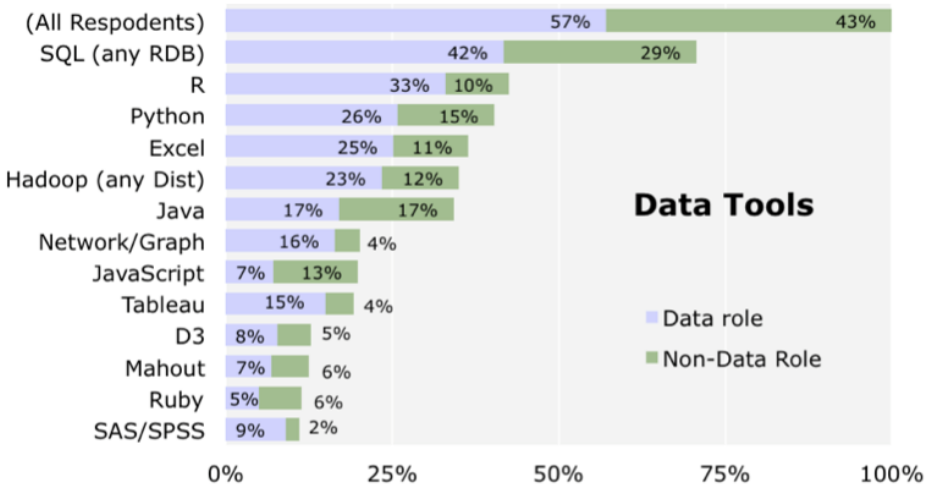
- Formát dát
- Vnútorná štruktúra dát
- Typ dát
- Objem dát
- Dostupnosť dát
- Väzby na sledované štatistické údaje

Definuje sa proces ich získavania, uskladnenia a čistenia pre účely analýzy.

Analyticky Sandbox slúži na prvotnú prácu s dátami predtým ako sa prejde do slučky č.2 (definícia analytických nástrojov a modelov).

Slučka č.2.

Definuje vhodný analytický nástroj a model na prácu s pripraveným dátovým súborom. Preferovaným nástrojom pre oblasť štatistiky je programovací jazyk R , existuje široká škála analytických nástrojov ako napr. Tableau , Mahout alebo Matlab. Zaujímavý je aj pohľad na prieskum od O'Reilly na obľúbenosť jednotlivých nástrojov na analýzu dát *(obr.).*



Pri výbere nástroja je nutné myslieť na jeho vhodnosť a opodstatnenosť voči hypotéze. Paralelne s výberom nástroja prebieha aj definícia vhodného analytického modelu.

Príloha č.2

Version 2.04_03.03.2014

Obsah

Úvod

Cieľom bolo otestovať sledovanie a zber neštruktúrovaných dát. Definícia jednoduchých nástrojov na analýzu textu. Práca v R ako nástroja na analýzu dát.

Záverečnú časť tvorí návrh tém na spracovanie a analýzu Big Data v prostredí SR.

Pojmy použité v dokumente

R - programovací jazyk a prostredie určené pre dátovú a štatistickú analýzu a jej grafické zobrazenie.

Text Mining (TM) - získavania informácií z textu na základe zadefinovaných parametrov. Text Mining zvyčajne zahŕňa proces štruktúrovania vstupného textu, definíciu väzieb a významu v rámci štruktúrovaných dát, hodnotenie a interpretácia výstupu. Medzi typické úlohy Test Mining patrí textová kategorizácia, zlučovanie textu, extrakcia konceptu entity/sledovaného textu, tvorba taxonómie, analýza sentimentu, sumarizácia dokumentov a modelovanie vzťahov medzi sledovanými entitami.

Twitter – sociálna sieť, ktorá umožňujúce svojim užívateľom posielať a čítať správy ostatných používateľov, tzv. *tweets* (*tweety*). Tweety sú textové príspevky do 140 znakov zobrazených na užívateľskom profile.

API - Application Programming Interface (API) - Definuje vzájomnú komunikáciu jednotlivých software-ových komponentov. Okrem prístupu k DB a počítačovému HW sa používa aj pre prácu v GUI (grafickom rozhraní).

Zber dát

Pre zber dát bol vybratý ako zdroj internet a dáta generované človekom (konkrétne **1100** a **1200** podľa členenia UNECE Divízie Štatistiky - Projektový team „Big Data“, Jún 2013). Pričom boli zvážené všetky ostatne možnosti *(viď. členenie)* ,ale ich dostupnosť je veľmi limitovaná.

Členenie podľa UNECE Divízia Štatistiky - Projektový team „Big Data“, Jún 2013

1. Dáta generované človekom: väčšinou neštruktúrované dáta.
    - ***1100***. *Sociálne siete: Facebook, Twitter, Tumblr atď.*
    - ***1200***. *Blog a komentáre*
    - 1300. Osobné dokumenty
    - 1400. Obrázky: Instagram, Flickr, Picasa atď.
    - 1500. Videá: Youtube atď.
    - 1600. Vyhľadávania na internete
    - 1700. Mobilné dáta: textové správy
    - 1800. Užívateľmi generované mapy
    - 1900. E-Mail

2. Tradičné obchodné systémy: väčšinou vysoko štruktúrované, zahŕňajúce transakcie, referenčné tabuľky a vzťahy, rovnako ako aj metadáta.
    - 21. Dáta produkované verejnou správou
        - 2110. Zdravotné záznamy
    - 22. Dáta produkované podnikateľským prostredím
        - 2210. Komerčné transakcie
        - 2220. Bankové záznamy/ Finančné trhy
        - 2230. E-commerce
        - 2240. Kreditné karty

3.Strojom generované dáta (Internet of Things) : Zvyčajne štruktúrované a pološtruktúrovane, časť uložená RDBMS.
    - 31. Dáta zo senzorov
        - 311. Fixné/pevné senzory
            - 3111. V domácnostiach
            - 3112. Na sledovanie počasia a znečistenia
            - 3113. Na sledovanie dopravy (vrátane foto-video techniky)
            - 3114. Vo vede a výskume
            - 3115. Bezpečnostné (vrátane foto-video techniky)
        - 312. Mobilné senzory (monitorovacie)
            - 3121. Mobilné zariadenia (priestorové súradnice)
            - 3122. Dopravné prostriedky
            - 3123. Satelitné snímky
    - 32. Dáta z počítačových systémov
        - 3210. Logs
        - 3220. Web logs

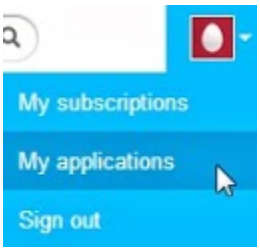
Bolo testovaných viacero internetových zdrojov (blogy, komentáre a iné dáta generované človekom) pre malú prípadovú štúdiu v tomto dokumente bol vybraný Twitter najme kvôli užívateľsky veľmi prijateľnému API. Množstvu neštruktúrovaných dát ,ktoré okrem písmen obsahujú aj čísla a #Hashtag, ktorý sa používa na opis daného tweetu. Jedná sa o frázu, ktorá začína znakom #. Twitter

premení hashtag na odkaz, pomocou ktorého je možné nájsť iné tweety s rovnakým hashtagom. Tieto hashtagy môžu označovať rôzne zaujímavosti (film, osobnosť, dôležitá udalosť) alebo označovať význam tweetu (radosť, hnev, smútok, sarkazmus) *(zdroj.Twitter).* Príklad: #infostat, #susr, #volby. Ďalej @ Zavináč, za znakom @ nasleduje meno užívateľa, znamená to, že daný užívateľ bol v konkrétnom príspevku spomenutý, alebo mu bolo odpovedané.

Twitter samozrejme podlieha limitom na množstvo stiahnutých odkazov čo samotný zber trocha obmedzuje najmä pri zbere väčšieho objemu odkazov *(Twitter limity: https://dev.twitter.com/docs/rate-limiting/1.1).*

Príklad sťahovanie live komentárov z Twitter-u na kľúčové slovo **#Slovakia** počas Zimnej Olympiády v Soči (počas hokejového zápasu Slovenska a Ruska). Dáta boli sťahované priamo do R bez použitia DB (NoSQL).

Na začiatok je potrebné na sieti Twitter https://dev.twitter.com/ v rámci vlastného užívateľského účtu vytvoriť aplikáciu na komunikáciu s cieľovým zariadením.



Následne sa zabezpečí certifikácia na komunikáciu s Twitter-om, nainštalujú sa potrebne pracovné rozhrania:

Skrátená verzia postupu priamo v R

```
# Load the required R libraries
#
> library(twitteR)
> library(ROAuth)
> library(RCurl)

#curl Cert
> download.file(url="http://curl.haxx.se/ca/cacert.pem",
destfile="cacert.pem")
The downloaded binary packages are in
        C:\Users\Suja\AppData\Local\Temp\RtmpOAoF9w\downloaded_packages
```

Nastavenie certifikácie na sieti Twitter a v R

Twitter: https://dev.twitter.com/docs/auth/oauth

v R:

```
> install.packages("C:/Users/Suja/Downloads/ROAuth_0.9.3.zip", repos = NULL)
Installing package into 'C:/Users/Suja/Documents/R/win-library/3.0'
(as 'lib' is unspecified)

package 'ROAuth' successfully unpacked and MD5 sums checked

> library("ROAuth", lib.loc="C:/Users/Suja/Documents/R/win-library/3.0")
Loading required package: RCurl
```

```
Loading required package: bitops
Loading required package: digest


> install.packages("C:/Users/Suja/Downloads/rjson_0.2.13.zip", repos = NULL)
Installing package into 'C:/Users/Suja/Documents/R/win-library/3.0'
(as 'lib' is unspecified)

package 'rjson' successfully unpacked and MD5 sums checked
```

Pripojenie na siet Twitter:

```
> requestURL <- "https://api.twitter.com/oauth/request_token"
> accessURL <- "https://api.twitter.com/oauth/access_token"
> authURL <- https://api.twitter.com/oauth/authorize

> consumerKey <- "HOk4DmSwLLxXT28W45EJA"
> consumerSecret <- "hR5M1FuXzLcNAi44ov9F3ZJThfSLIm5WqxKWsk30"

> twitCred <- OAuthFactory$new(consumerKey=consumerKey,
+                     consumerSecret=consumerSecret,
+                     requestURL=requestURL,
+                     accessURL=accessURL,
+                     authURL=authURL)
> twitCred$handshake(cainfo="cacert.pem")
To enable the connection, please direct your web browser to:
https://api.twitter.com/oauth/authorize?oauth_token=WlzAdtWfnzg7VWt9VbgbQrr
nPWhfjsVPw7eNpiu9A
When complete, record the PIN given to you and provide it here: 2758282

> registerTwitterOAuth(twitCred)

[1] TRUE
```

Sťahovanie dát zo siete Twitter:

```
> hokej.sk <- searchTwitter('#Slovakia', cainfo="cacert.pem")
```

Pohľad na stiahnuté dáta:

```
> head ("hokej.sk", n=12L)

    [[1]]
    [1] "nancymenagh: #czechrepublic vs. #slovakia #icehockey #hock
    ey #sochi #olympics http://t.co/lJkpBZy45m"

    [[2]]
    [1] "NewsDetector: US crushes Slovakia 7-1 in men's Olympic hoc
    key http://t.co/wsfNlvBtjk\n #JaroslavHalak #NHL #Russia #slova
    kia #ZdenoChara"

    [[3]]
    [1] "MiaRadke: Czech Republic VS Slovakia...alright let's do it
    ... #Sochi2014  #Slovakia"

    [[4]]
    [1] "MarkLazerus: #Slovakia #CzechRepublic #Go"

    [[5]]
    [1] "MartinJaniSK: RT @spodlesny: Za chvĂlu zaÄŤĂname. Ideme vy
    bojovaĹĄ ÄŚechom letenku domov :) #Slovensko #Slovakia #hokej"

    [[6]
```

```
[1] "Tina_Ivanko: Uf! ..lets go guys..so hard to keep calm :D #
goslovakia #teamslovakia #slovakia #vs #czechrepublic #olympics
#Sochi2014"
[[7]]
[1] "Blackyn92: Lets go #Slovakia  ! 0-2 down but nothing is lo
st yet.  #icehockey #Olympics2014"

[[8]]
[1] "TomCroke3: Pavelec looked mad that the net came loose. #Sl
ovakia #CzechRepublic"

[[9]]
[1] "fritz114: When you can't score, just run the goalie. #Slov
akia #Czechs #Sochi2014"

[[10]]
[1] "NHLNewsPuck: Preview: #CzechRepublic vs. #Slovakia ..#Flam
es #NHLNews http://t.co/9ZJE9vfZXQ"

[[11]]
[1] "dawnyyyk: Wouldn't be an international tourney if it weren
't for upsets. Getting depressed right about now... #Slovakia #
Switzerland #Leggo"

[[12]]
[1] "BoabyLuv16: Horrible start for #Slovakia. #Sochi2014"
```

V tejto chvíli sú dáta pripravene na ďalšie spracovanie (čistenie a triedenie) a následnú analýzu.

Analýza neštruktúrovaných dát

Ďalším krokom po zbere dát je samotná analýza. V našom prípade neštruktúrovaných dát – textu. Vzhľadom na limity siete Twitter, ktoré boli spomenuté v predchádzajúcej kapitole a navyše k nízkemu počtu textových záznamov na kľúčové slova ako Slovakia, Slovensko atď.. Bol pre účel prípadovej štúdie vybraný iný súbor a to zdokumentované záznamy o videní UFO - v objeme 62 476 riadkov vo formáte json. Uvedený súbor je priložený k nahliadnutiu v dvoch formátoch (json a tsv).


ufo_awesome.json


ufo_awesome.tsv

Príklad súboru json *(2 riadky)*:

{"sighted_at": "19951009", "reported_at": "19951009", "location": " Iowa City, IA", "shape": "", "duration": "", "description": "Man repts. witnessing "flash, followed by a classic UFO, w/ a tailfin at back." Red color on top half of tailfin. Became triangular."}{"sighted_at": "19951010", "reported_at": "19951011", "location": " Milwaukee, WI", "shape": "", "duration": "2 min.", "description": "Man on Hwy 43 SW of Milwaukee sees large, bright blue light streak by his car, descend, turn, cross road ahead, strobe. Bizarre!"}{"sighted_at": "19950101", "reported_at": "19950103", "location": " Shelton, WA", "shape": "", "duration": "", "description": "Telephoned Report:CA woman visiting daughter witness discs and triangular ships over Squaxin Island in Puget Sound. Dramatic. Written report, with illustrations, submitted to NUFORC."}

Samotný proces analýzy sa dá rozdeliť do 2 krokov:

1. Čistenie a triedenie dát
2. Samotná analýza dát

Všetky úkony boli vykonané v R-ku (použité boli balíky – tm, Rweka, wordcloud, Rccp, ggplot2)

Z dôvodu technických obmedzení bola nakoniec zo súboru vybraná iba vzorka pre účely textovej analýzy.

Členenie vytriedených a očistených dát pred odberom vzorky:

```
Dátum Zjavenia      Dátum Reportovania        Lokalita
Krátky popis     Trvanie zjavenia   Obsiahly popis
```

Jednoduchá analýza dátového súboru:

```
  Dátum Zjavenia      Dátum Reportovania       Lokalita
Min.   :1400-06-30   Min.   :1905-06-23   Seattle, WA    :  444
1st Qu.:1999-09-15   1st Qu.:2002-05-08   Phoenix, AZ    :  378
Median :2003-12-15   Median :2005-02-27   Los Angeles, CA:  297
Mean   :2001-02-10   Mean   :2004-11-22   nezadefinovana :  279
3rd Qu.:2007-06-21   3rd Qu.:2007-12-20   San Diego, CA  :  274
Max.   :2010-08-30   Max.   :2010-08-30   Las Vegas, NV  :  272
NA's   :733          NA's   :477          (Other)        :59926
```

```
  Krátky popis     Trvanie zjavenia
  light   :12202   nezadefin.: 2759
  triangle: 6082   5 minutes : 2375
  circle  : 5271   2 minutes : 1751
  disk    : 4825   1 minute  : 1741
  other   : 4593   10 minutes: 1698
  unknown : 4490   (Other)   :51543
 (Other)  :24407   NA's      :    3
```

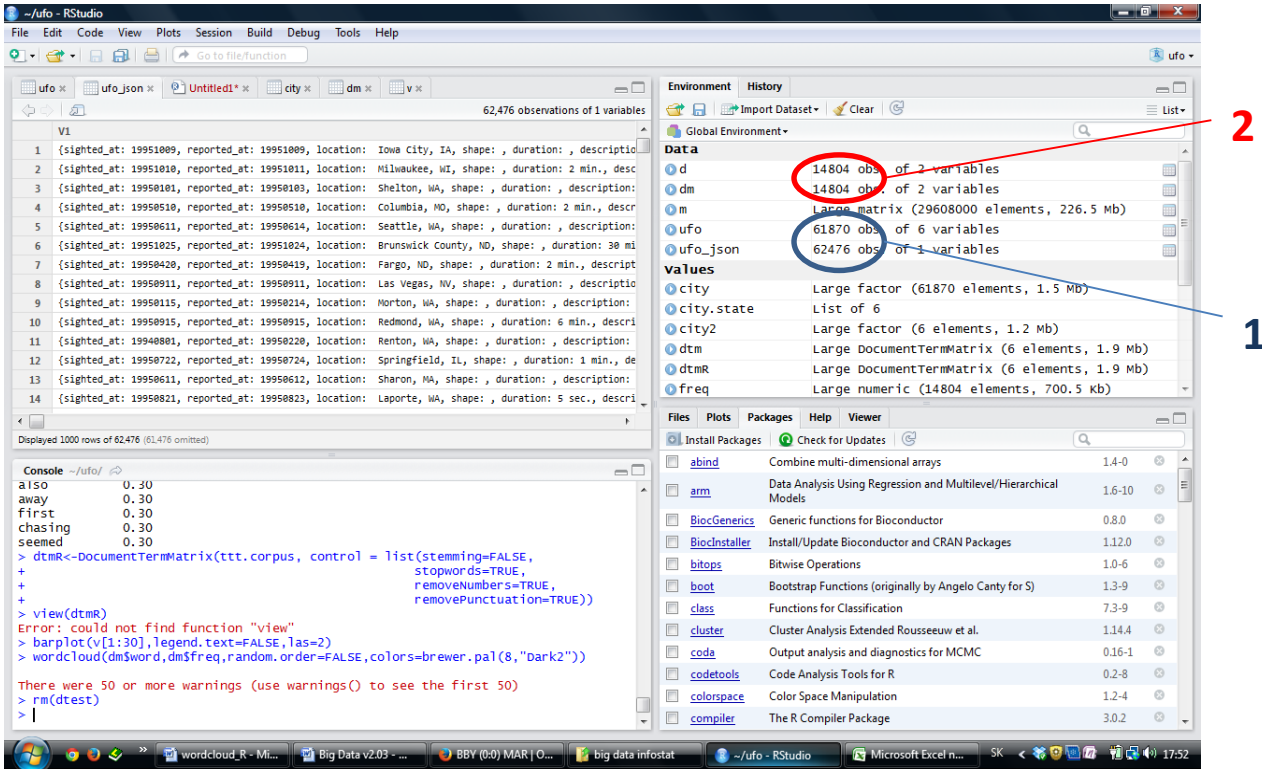
Najčastejšie miesta výskytu UFO:

Poradie	Mesto	Počet výskytov
1	Seattle, WA	444
2	Phoenix, AZ	378
3	Los Angeles, CA	297
4	neznama lokalita	279
5	San Diego, CA	274
6	Las Vegas, NV	272
7	Portland, OR	254
8	Houston, TX	248
9	Chicago, IL	220
10	Tucson, AZ	187
11	London (UK/England),	174
12	Miami, FL	162
13	Austin, TX	159

14	San Francisco, CA	153
15	San Antonio, TX	151
16	Toronto (Canada), ON	144
17	Albuquerque, NM	139
18	Denver, CO	137
19	Orlando, FL	137
20	Sacramento, CA	135
21	Tinley Park, IL	135
22	San Jose, CA	134
23	New York City, NY	123
24	Dallas, TX	115
25	Charlotte, NC	109
26	Mesa, AZ	106
27	Tacoma, WA	104
28	Jacksonville, FL	102
29	Rockford, IL	98
30	Spokane, WA	98

Zo Slovenských miest sa v súbore vyskytli Prievidza a Ivánka pri Dunaji.

Parametre vzorky pre textovú analýzu: prvých 600 riadkov z celkového súboru, ktorý sa pre účely textovej analýzy ďalej prefiltroval iba na položku „Obsiahly popis“ z čoho následne vznikol súbor o počte 14 804 riadkov (kde jeden riadok predstavoval jedno slovo a frekvenciu jeho výskytu – počet). Porovnanie pôvodného súboru 1 a vzorky 2 na obrázku.



Samotná analýza spočívala najme v zisťovaní frekvencie výskytu používania daného slova v popise udalosti a zisťovaní vzťahu medzi nimi (korelácie).

Grafické znázornenie frekvencie používania jednotlivých slov *(300 najčastejšie používaných slov) :*

Korelácia medzi jednotlivými slovami, napríklad slovo „light“ vo vzťahu k iným slovám:

	light
car	0.36
towards	0.36
bright	0.35
continued	0.34
like	0.33
see	0.33
seen	0.33
thought	0.33
back	0.32
drove	0.32
just	0.32
now	0.32
saw	0.32
summary	0.32
time	0.32
considerably	0.31
even	0.31
far	0.31
grasp	0.31
two	0.31
white	0.31
also	0.30
away	0.30
first	0.30
chasing	0.30
seemed	0.30

Iný príklad - slovo „UFO“ je uvedený v prílohe.

Príloha

	ufo
ancestors	0.72
annoyed	0.72
center	0.72
clara	0.72
command	0.72
concern on	0.72
courts	0.72
crawl	0.72
dear	0.72
expressway	0.72
goquot	0.72
granada	0.72
grayishwhite	0.72
hourto	0.72
ideas	0.72
incorrectly	0.72
indefinable	0.72
kwong	0.72
lesson	0.72
lightening	0.72
manmade	0.72
metallic red	0.72
metallic silverblue	0.72
minimum	0.72
people end	0.72
peteralong	0.72
play	0.72
practicing	0.72
quottrackquot	0.72
report	0.72
retell	0.72
serve	0.72
skimmed	0.72
station seattle	0.72
testimony	0.72
thorough	0.72
toss	0.72
twelveyearsold	0.72
yearsold	0.72
teaching	0.70
michael	0.69
teach	0.69
ufoaposs	0.68
tennis	0.67
yearold	0.59
post	0.58
anyway	0.55
grandmother	0.54
transfer	0.53
court	0.51
enclosed	0.51
handwritten	0.51
harder	0.51
lone	0.51
marina	0.51
necks	0.51
react	0.51
share	0.51
silver	0.51
silverblue	0.51
speaking	0.51
switched	0.51
tossing	0.51
jets	0.47
internet	0.46
settled	0.45
completely	0.44
jose	0.44
peter	0.44
accounting	0.43
story	0.43
fence	0.42
sightings	0.42
emailed	0.41
metallic	0.41
motheraposs	0.41
paths	0.41
resumed	0.41
saturday	0.41
writes	0.41
amazing	0.40
half	0.40
reporting	0.40
take	0.40
writing	0.40
credible	0.39
curved	0.38
iaposll	0.38
whitish	0.38
hour	0.37
national	0.37
red	0.37
atmosphere	0.36
difficulty	0.36
etc	0.36
lives	0.36
apt	0.35
entering	0.35
exciting	0.35
parted	0.35
santa	0.35
saw	0.35
sighting	0.35
someone	0.35
unlike	0.35
another	0.34
believe	0.34
engineer	0.34
personnel	0.34
really	0.34
tracking	0.34
air	0.33
altitude	0.33
life	0.33
amazement	0.32
given	0.32
indoors	0.32
least	0.32
much	0.32
related	0.32
report	0.32
slowly	0.32
tired	0.32
always	0.31
goes	0.31
orangered	0.31
skeptical	0.31
find	0.30
force	0.30
jet	0.30
passenger	0.30
told	0.30

Príloha č.3

Version 3.04_08.11.2014

Softwarové riešenie pre webscraping (program, ktorý automaticky prejde webstránky a vykoná zápis nálezu do csv súboru) naprogramované v jazyku PERL je priložené na koniec tohto textu.

Postupnosť krokov pre spustenie programu:

1. Testovaný súbor SSJ_DEF.xls obsahuje 4247 jednotiek (dodaný ŠÚSR)

2. Manuálna filtrácia súboru - odstránene sú všetky jednotky, ktoré neobsahujú položku email alebo štatistické jednotky používajú voľne dostupne emailové schránky ako napr. gmail, zoznam, post, yahoo atď.

3. Zvyšné emailové adresy sú transformované na web adresy -> výsledný súbor - jednotky_db.xlsx

4. Definícia sledovaných kritérií pre webscraping. Na základe otázok 13-24 dotazníka Zisťovanie o informačných a komunikačných technológiách (IKT) v podnikoch, boli definovane hlavne tematické okruhy:

    a. Webstránka obsahuje eshop?

    b. Webstránka obsahuje linky na sociálne siete ako facebook alebo twitter?

    c. Webstránka obsahuje certifikát bezpečnosti (SSL) - https

    d. Webstránka obsahuje multimédia ako video alebo hudbu

    e. Na webstránke sa nachádzajú kontaktne údaje - email a telefón spoločnosti

5. Definícia hodnotiacich kritérií pre webscraping:

    a. Reálna existencia webstránky (nie všetky vyfiltrovane webstránky existujú napr. email - suja@suja.sk<mailto:suja@suja.sk> nezaručuje existenciu www.suja.sk<http://www.suja.sk>). Hodnotenie: pokiaľ stránka existuje zápis do výsledného csv súboru je 1 pokiaľ nie 0

    b. Má stránka eshop: 1-ano 0-nie

    c. Má stránka linky na sociálne siete: 1-ano 0-nie

    d. Má stránka certifikát bezpečnosti: 1-ano 0-nie

    e. Má stránka multimédia: 1-ano 0-nie

    f. Pokiaľ stránka obsahuje kontaktne údaje email a telefón tie sú extrahované do výsledného csv súboru.

6. Webscraping nastroj - Perl script na konci tohto textu: WebScraping_Suja_Infostat

7. Spustenie "WebScraping_Suja_Infostat.pl"

a. Potrebná inštalácia ActiveState perl - http://www.activestate.com/activeperl/downloads

b. Otvorenie "package manager" a inštalácia LWP::UserAgent (http://search.cpan.org/dist/LWP-Protocol-https-6.04/)

c. V programe je potrebné definovať cestu k súboru z ktorého chceme načítať dáta a zapísať nové (v programe sa nahradí "path_to_file" novou nadefinovanou cestou)

d. Môžeme pustiť program"WebScraping_Suja_Infostat.pl" z príkazového riadka

8. Výsledky sú zapísané do csv súboru "all_in_one_jednotky_db" vid. príloha

9. csv súbor je pripravený na analýzu

Perl Script pre WebScraping:

```
################################################################################################
# Robert Suja
# Infostat 2014
# Big Data Project - WebScraping
################################################################################################

use strict;
use warnings;
use LWP::UserAgent;
use HTTP::Request::Common qw(GET);
use HTTP::Cookies;

my $ua = LWP::UserAgent->new;

# Define user agent type
$ua->agent('Mozilla/8.0');

################################################################################################
################ Novy subor pre zapis vysledkov
################################################################################################

my $outfile = 'path_to_file\all_in_one_jednotky_db_csv.csv';
open(my $out_fh, ">", $outfile) or die "Could not open file '$outfile' $!";

my @lines ;
my %ResultHash ;

################################################################################################
################## Nacitaj csv file
################################################################################################

my $filename = 'path_to_file\jednotky_db_csv.csv';
open(INFILE, '<', $filename) or die "Could not open file '$filename' $!";

my @weblinks ;

print $out_fh "ICO;NAZ23;web
adresa;KOR_NAZ;KOR_UL;KOR_CIS;KOR_PSC;KOR_OBEC;KOR_POSTA;WEB PAGE
EXISTS;SHOP;FACEBOOK;TWITTER;VIDEO\/MUSIC;SSL;MAIL;PHONE\n";

my $i = 0 ;

#### Sprocesuj kazdu webstranku
```

```
while (my $row = <INFILE>) {
 chomp $row;
 @weblinks = split(';',$row);
 if ( $i > 0 ) {
        # Check Site
        getInfo($weblinks[2]);
        print "---------- WEBLINK $weblinks[2]\n";
        print "EXISTS $ResultHash{$weblinks[2]}{'exists'}\n";
        print "SHOP $ResultHash{$weblinks[2]}{'Shop'}\n";
        print "FACEBOOK $ResultHash{$weblinks[2]}{'facebook.com'}\n";
        print "TWITTER $ResultHash{$weblinks[2]}{'twitter.com'}\n";
        print "AUDIO-VIDEO $ResultHash{$weblinks[2]}{'audio-video'}\n";
        print "SSL $ResultHash{$weblinks[2]}{'ssl'}\n";
        print "EMAIL $ResultHash{$weblinks[2]}{'email'}\n";
        print "PHONE $ResultHash{$weblinks[2]}{'phone'}\n";
        print $out_fh
"$weblinks[0];$weblinks[1];$weblinks[2];$weblinks[3];$weblinks[4];$weblinks[5];$weblinks[6];$webl
inks[7];$weblinks[8];$ResultHash{$weblinks[2]}{'exists'};$ResultHash{$weblinks[2]}{'Shop'};$ResultH
ash{$weblinks[2]}{'facebook.com'};$ResultHash{$weblinks[2]}{'twitter.com'};$ResultHash{$weblinks[
2]}{'audio-
video'};$ResultHash{$weblinks[2]}{'ssl'};$ResultHash{$weblinks[2]}{'email'};$ResultHash{$weblinks[2
]}{'phone'}\n";
        $out_fh->flush();
 }
  $i++ ;
}

close $out_fh;

##########################################################################################
################## "Parse" vysledkov.
##########################################################################################

sub getInfo {
        my $link = shift ;

        my $sua = LWP::UserAgent->new(ssl_opts => { verify_hostname => 1 });
        my $sres = $ua->get("https://$link");

        $ResultHash{$link}{'ssl'}=0;

        if ($sres->is_success) {
                $ResultHash{$link}{"ssl"}=1;
        }
```

```
my $req = HTTP::Request->new(GET => "http://$link");
my $res = $ua->request($req);
my $pageContent ;


if ($res->is_success) {
	$ResultHash{$link}{"exists"} = 1;
	$pageContent = $res->content;

	$ResultHash{$link}{"Shop"} = 0 ;
	if ( $pageContent =~ m/[Ss]hop/ ) {
		$ResultHash{$link}{"Shop"} = 1 ;
	} else {
		if ( $pageContent =~ m/obchod/ ) {
			$ResultHash{$link}{"Shop"} = 1 ;
		} else {
			$ResultHash{$link}{"Shop"} = 0 ;
		}
	}

	$ResultHash{$link}{"twitter.com"}=0;
	if ( $pageContent =~ m/twitter.com/ ) {
		$ResultHash{$link}{"twitter.com"} = 1 ;
	} else {
		$ResultHash{$link}{"twitter.com"} = 0 ;
	}

	$ResultHash{$link}{"facebook.com"} = 0 ;
	if ( $pageContent =~ m/facebook\.com/ ) {
		$ResultHash{$link}{"facebook.com"} = 1 ;
	} else {
		$ResultHash{$link}{"facebook.com"} = "0" ;
	}

	$ResultHash{$link}{"audio-video"} = 0 ;

	if ( $pageContent =~ m/\.mp4/ ) {
		$ResultHash{$link}{"audio-video"} = 1 ;
		print "mp4 exists \n";
	} elsif ( $pageContent =~ m/\.mpg/ ) {
		$ResultHash{$link}{"audio-video"} = 1 ;
		print "mpg exists \n";
	} elsif ( $pageContent =~ m/\.mp3/ ) {
		$ResultHash{$link}{"audio-video"} = 1 ;
		print "mp3 exists \n";
```

```
		} elsif ( $pageContent =~ m/\.avi/ ) {
			$ResultHash{$link}{"audio-video"} = 1 ;
			print "avi exists \n";
		} elsif ( $pageContent =~ m/\.wav/ ) {
			$ResultHash{$link}{"audio-video"} = 1 ;
			print "wav exists \n";
		} elsif ( $pageContent =~ m/rtmp\:/ ) {
			$ResultHash{$link}{"audio-video"} = 1 ;
			print "rtmp exists \n";
		} else {
			$ResultHash{$link}{"audio-video"} = 0 ;
		}

		$ResultHash{$link}{"email"} = "";
		if ( my @emails = $pageContent =~
m/["'\n\r\s\t\>\<\(\)\/]([^"'\n\r\s\t\>\<\(\)\/]+\@[^"'\n\r\s\t\>\<\(\)\/]+.[^"'\n\r\s\t\>\<\(\)\/]+)["'\
n\r\s\t\<\>\(\)\/]/g ) {
			foreach my $email (@emails) {
				$email =~ s/mailto:(.+)/$1/g ;
				chomp($email);
				$ResultHash{$link}{"email"} =
$ResultHash{$link}{"email"} . ' | ' . $email . ' | ' ;
			}
			print "MAILS: $ResultHash{$link}{'email'}\n";

		}

		$ResultHash{$link}{"phone"} = "";
		if ( my @phones = $pageContent =~
m/["'\n\r\s\t\<\>]([\d\(\)][\d\s\(\)]{7,12}[\d\(\)])["'\n\r\s\t\<\>]/g ) {
			foreach my $phone (@phones) {
				chomp($phone);
				$ResultHash{$link}{"phone"} =
$ResultHash{$link}{"phone"} . ' | ' . $phone . ' | ' ;
			}
			print "$ResultHash{$link}{'phone'}\n";
		}
	} else {
		$ResultHash{$link}{"exists"} = 0;
		$ResultHash{$link}{"Shop"} = 0 ;
		$ResultHash{$link}{"twitter.com"}=0;
		$ResultHash{$link}{"facebook.com"}=0;
		$ResultHash{$link}{"audio-video"} = 0 ;
		$ResultHash{$link}{"email"} = "";
		$ResultHash{$link}{"phone"} = "";
```

```
                print "$link not succesfull \n" ;
                #print $res->status_line, "No such line\n";
        }

}
```