NÁSKOK
DÍKY
ZNALOSTEM

PROFINIT

NDBI047
Aplikace bigdat v Data Science

Storage, Hive

Milan Kratochvíl, Jan Hučín

8. 3. 2019

Osnova přednášky

- Hadoop a HDFS – opakování
- Formáty ukládání dat v Hadoopu
- Komprese
- Hive
- Shrnutí

PROFINIT

Hadoop a HDFS – opakování

Hadoop – opakování

- distribuované ukládání
- distribuované zpracování
- velké soubory

- Důležité komponenty (pro nás):
  - HDFS
  - Hive (Impala)
  - MapReduce
  - Spark

PROFINIT

HDFS – opakování

- distribuovaný file system
- Namenode vs Datanode
- replikace (3 by default)
- uživatel nevidí, kde je který soubor fyzicky
- preferovány velké soubory

PROFINIT

Formáty pro ukládání

Typy souborů

- Binární
  - exporty zdrojových systémů (CDRs - telco)
  - videa, obrázky, zvukové záznamy
- Textové
  - prostý text
  - CSV
  - XML
  - JSON

PROFINIT

Formáty pro ukládání

- Binární – beze změny
- Prostý text – beze změny
- „Tabulární“ řádkově orientované úložiště
  - **CSV** – nenese informaci o schématu
  - **Avro** (https://avro.apache.org/) – ve skutečnosti nástroj na serializaci, obsahuje schéma
- „Tabulární“ sloupcově orientované úložiště – v metadatech obsaženo schéma
  - **RCfile**
  - **ORC** (https://orc.apache.org/)
  - **Parquet** (https://parquet.apache.org/)
- Ostatní
  - **SequenceFile**

SequenceFile

- Řešení problému s mnoha malými soubory
- Umožňuje ukládat data ve formátu key/value
  (key: název souboru, value: obsah souboru)
- Umožňuje blokové rozdělení/blokovou kompresi
- Umožňuje přidávání dat

Řádkově a sloupcově orientované úložiště

RowId	EmpId	Lastname	Firstname	Salary
001	10	Smith	Joe	40000
002	12	Jones	Mary	50000
003	11	Johnson	Cathy	44000
004	22	Jones	Bob	55000

- přístup po řádcích – mazání, úprava
- přístup po sloupcích – analýza
- Reprezentace úložiště

řádkově

```
001:10,Smith,Joe,40000;
002:12,Jones,Mary,50000;
003:11,Johnson,Cathy,44000;
004:22,Jones,Bob,55000;
```

sloupcově

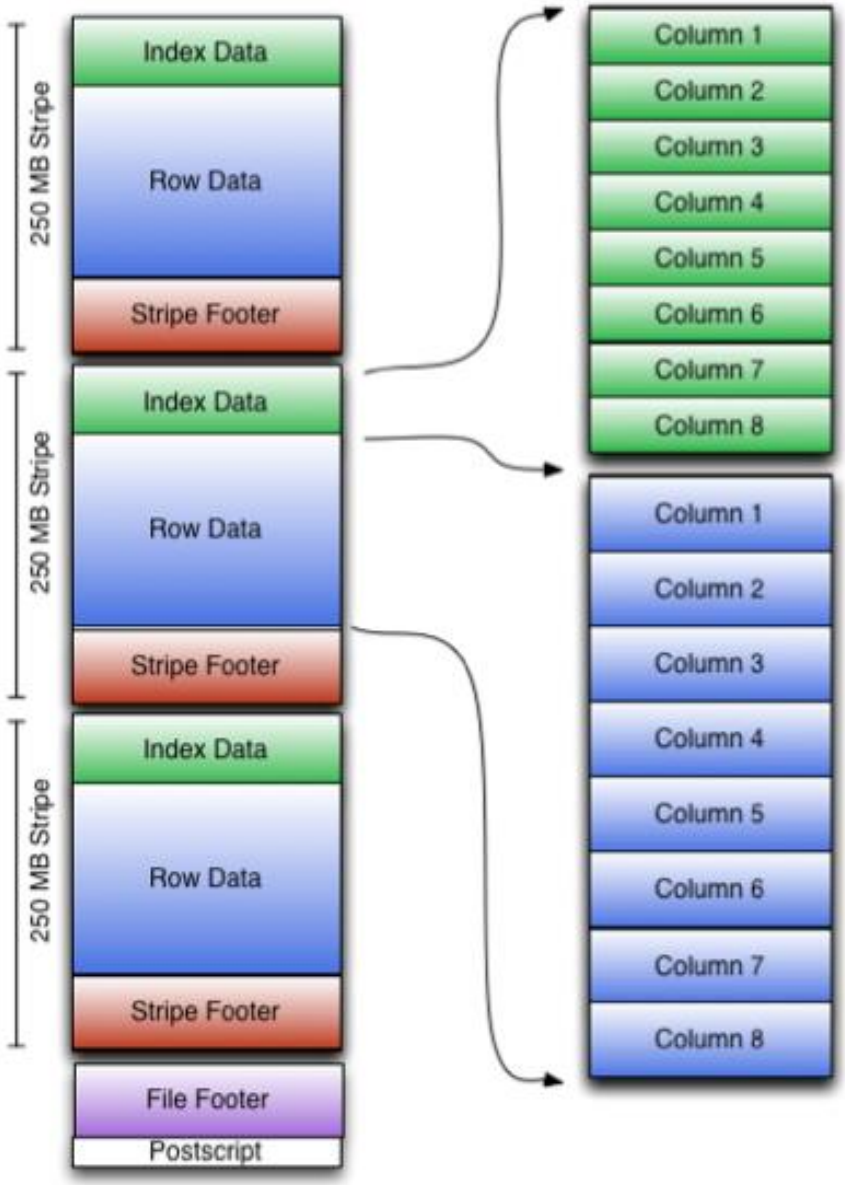
```
10:001,12:002,11:003,22:004;
Smith:001,Jones:002,Johnson:003,Jones:004;
Joe:001,Mary:002,Cathy:003,Bob:004;
40000:001,50000:002,44000:003,55000:004;
```

Sloupcově orientované úložiště – pro a proti

- Výhody
  - Možnost **rychle** číst **jen sloupce,** které potřebuji
  - Efektivnější komprese
  - Skvělé pro OLAP použití
- Nevýhody
  - Práce s jedním záznamem a full scan jsou pomalé
  - Nelze snadno modifikovat záznam
  - Náročný pro zápis
    - Vstupní data → bloky řádků → po sloupcích
    - Blok řádků se musí vejít do bloku HDFS
  - Špatné pro OLTP použití

PROFINIT

Příklad: ORC


250 MB Stripe
Index Data
Row Data
Stripe Footer
250 MB Stripe
Index Data
Row Data
Stripe Footer
250 MB Stripe
Index Data
Row Data
Stripe Footer
File Footer
Postscript
Column 1
Column 2
Column 3
Column 4
Column 5
Column 6
Column 7
Column 8
Column 1
Column 2
Column 3
Column 4
Column 5
Column 6
Column 7
Column 8

Sloupcově orientovaná úložiště – druhy

- RCFile – jen historické aplikace
- ORC – Optimized Row Columnar file format
  - vše, co RCFile + něco navíc
  - ukládá metadata/statistiky (average, max, count...)
  - může obsahovat i indexy
  - načítá jen potřebná data – optimalizuje dotaz
- Parquet – velmi podobné, jednodušší/lightweight

PROFINIT

Komprese

Komprese

Proč?

- Výrazně zrychlí přístup k datům

Jak?

- Gzip/Zlib
  - základní v Hadoopu
  - typicky s formátem ORC
- Bzip2
- LZO
- Snappy
  - typicky s formátem Parquet
  - nižší účinnost, ale nejrychlejší

Kompresní algoritmy - srovnání

Algoritmus	Rychlost	Účinnost	„Splittable“
GZIP/ZLib		✓	
BZip2		✓	✓
LZO	✓		✓
Snappy	✓		

- „Splitovatelnost“
  - kompresní algoritmus vytváří bloky, které lze samostatně dekomprimovat
  - vhodné pro paralelní zpracování
- Kompatibilita
  - **Ne každý formát a každý nástroj podporuje libovolnou kompresi!** (Např. v Impala nelze použít ORC)

PROFINIT

Hive

Hive

- https://hive.apache.org/
- Snaha přivést SQL do světa Hadoopu
- Nástroj pro dotazování a manipulaci s daty
- Vlastní jazyk HQL (variace na SQL)
- Exekuce dotazů – jiné komponenty (MapReduce, Tez, Spark)
- Poměrně vyspělý a mocný nástroj

Hive – ukládání dat

- DB tabulka = adresář
  - může obsahovat více souborů i další podadresáře
  - data se interpretují podle zadaného schématu
- Externí vs. interní tabulka
  - externí tabulka – link na existující adresář, vlastníkem je kdokoliv
  - managed (též interní) tabulka – adresář ve větvi HDFS vyhrazené pro Hive, vlastníkem je Hive
- Lze použít různé formáty dat i způsoby komprese
  - Parquet, Avro, CSV, ORC, ...
  - Gzip, Snappy, ...
- Metadata jsou uložena v Metastore
  - klasická relační DB – MySQL, PostgreSQL
  - umístění souborů, statistiky, práva

Hive – HQL

- DDL (Data Definition Language)
  - `CREATE [EXTERNAL] TABLE`
  - `DROP TABLE`
  - `TRUNCATE TABLE`
  - `ALTER TABLE`
- DML (Data Manipulation Language)
  - `LOAD DATA`
  - `INSERT INTO TABLE, INSERT OVERWRITE TABLE`
- Query
  - `SELECT`

- Nelze obecně
  - `UPDATE`
  - `DELETE`
  - *(až na specifické případy tabulek s podporou ACID)*

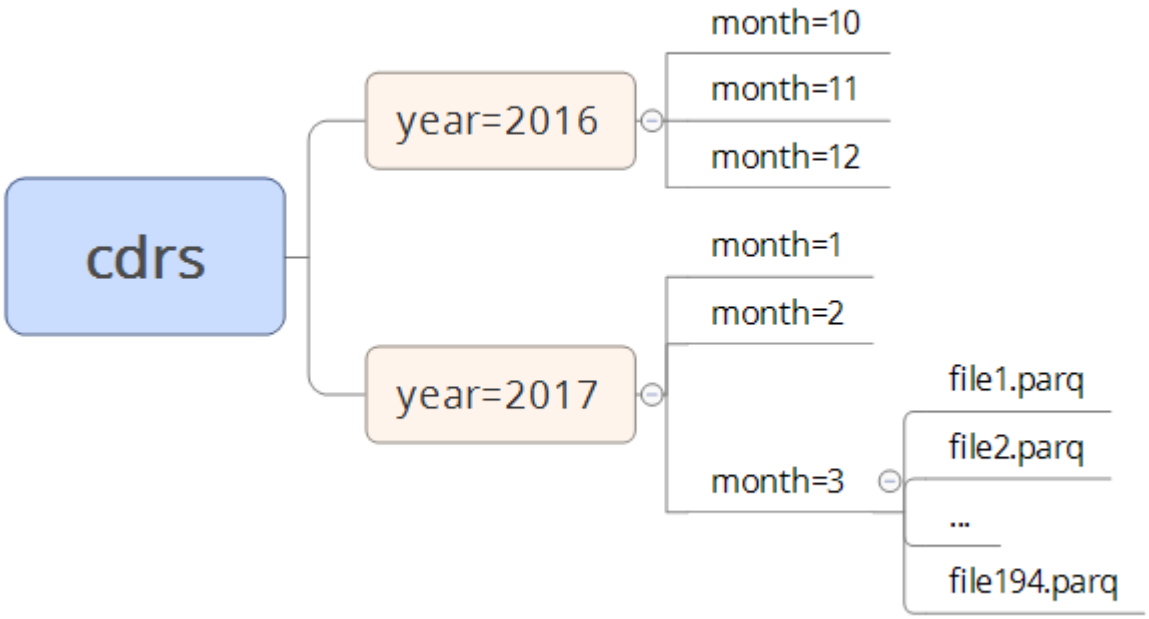
Loady a inserty dat

- 1 tabulka = 1 až N souborů
- **Každý INSERT vytvoří vždy aspoň jeden nový soubor!**
  - nedává tedy smysl INSERTovat záznam po záznamu
  - vždy INSERT z tabulky (INSERT SELECT, např. pomocí externí tabulky)

- Připomínka: mnoho malých souborů škodí

Partitioning (velmi důležité)

- Příklad:
  - Data chodí s časovým razítkem a jejich velmi mnoho (např. tel. hovory)
  - Typicky mě ale zajímají jen údaje za konkrétní den/měsíc...
- Partitioning
  - Logické rozdělení struktury tabulky do podadresářů
  - S každou lze pracovat samostatně
  - Dynamický (automatický) nebo statický (ruční) partitioning



PROFINIT

Hive – další možnosti

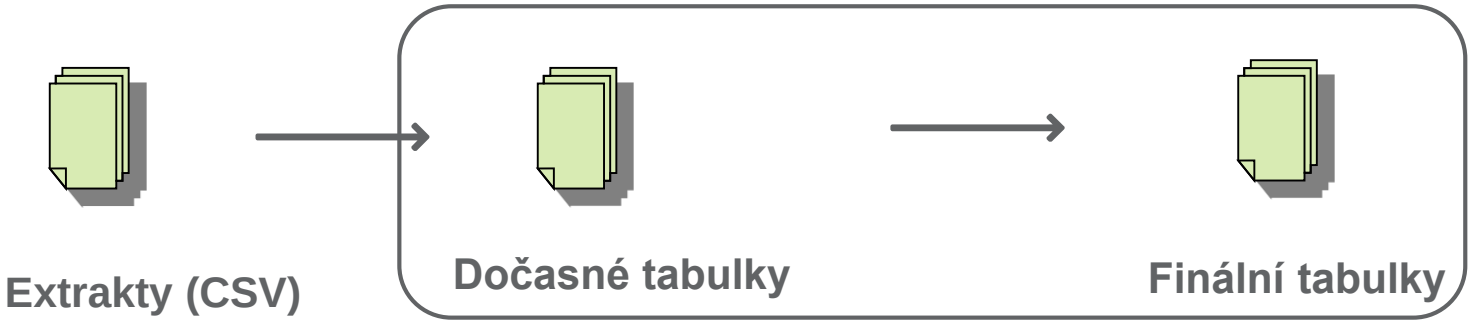
- Indexy
  - podpora pro indexy, možnost psát vlastní indexery
  - ale moc se nevyužívají – kvůli povaze HDFS
- Bucketing
  - rozděluje data do definovaného počtu „kyblíčků“ podle zvoleného sloupce
  - využití např. pro zrychlení JOINu
- Příliš mnoho partitions a bucketů může znamenat velmi malé soubory
- Při návrhu partitions/buckets je třeba být uvážlivý. Prakticky nelze změnit jinak než reloadem dat!

PROFINIT

Hive – Execution engine

- Původně se využíval MapReduce
  - pomalé
  - intenzivní zápisy na disk
  - ale paměťově nenáročné
- Dnes Hive umožňuje nastavit „execution engine“
  - MapReduce (Hive on MapReduce)
  - Tez (Hive on Tez)
  - Spark (Hive on Spark)

Hive – příklad



PROFINIT

Hive – příklad

- Vytvoříme externí tabulku vázanou na zdrojové soubory

```
CREATE EXTERNAL TABLE IF NOT EXISTS ap_temp (
    ACC_KEY BIGINT,
    PROD_ID VARCHAR(255),
    START_DATE TIMESTAMP,
    PROD_DESCR VARCHAR(255)
)
ROW FORMAT
    DELIMITED FIELDS TERMINATED BY ','
    LINES TERMINATED BY '\n'
STORED AS TEXTFILE
LOCATION '/data/input/acc';
```

Hive – příklad

- Vytvoříme prázdnou optimalizovanou tabulku

```
CREATE TABLE IF NOT EXISTS ap (
    ACC_KEY BIGINT,
    PROD_ID VARCHAR(255),
    START_DATE TIMESTAMP)
PARTITIONED BY (PROD_DESCR VARCHAR(255))
CLUSTERED BY (ACC_KEY) INTO 32 BUCKETS
STORED AS ORC tblproperties
    ("orc.compress"="ZLIB");
```

Hive – příklad

- Nahrajeme data do optimalizované tabulky

```
INSERT OVERWRITE TABLE ap
PARTITION (PROD_DESCR)
SELECT
 ACC_KEY,
 PROD_ID,
 START_DATE,
 PROD_DESCR
FROM ap_temp;


DROP TABLE ap_temp;
```

PROFINIT

Shrnutí
+ Co se jinam nevešlo

Schema evolution

- Změna schématu souboru v průběhu života
- Typicky chceme
  - přidat sloupec
- Někdy je i možné
  - přejmenovat sloupec
  - odstranit sloupec (zpravidla jen v metadatech)
- Podpora schema evolution záleží na formátu souboru i použitém nástroji – vždy je třeba nastudovat
- Podporované formáty alespoň pro přidávání sloupců
  - Avro
  - ORC
  - Parquet

Shrnutí

- Vždy se rozmýšlet, co je třeba, podle požadovaného použití
  - každý nástroj nabízí jinou míru flexibility!
- Snažit se maximálně těžit ze sloupcově orientovaných úložišť
  - např. analytické „SQL-like“ úlohy
- Použít řádkově orientované úložiště tam, kde je nutný full scan
- Programování (MapReduce) a SQL dotazování (Hive) se musí přizpůsobit světu Hadoopu
- Soubory v databázi Hive je třeba mít rozumně velké – ideálně velikost HDFS bloku (po kompresi)
- Partitioning je základ optimalizace v Hive
  - ale nepřehánět – je dobré mít zhruba odhad velikosti partition

Díky
za pozornost

PROFINIT
NÁSKOK DÍKY ZNALOSTEM

Profinit EU, s.r.o.
Tychonova 2, 160 00 Praha 6 | Telefon + 420 224 316 016

Web
www.profinit.eu

LinkedIn
linkedin.com/company/profinit

Twitter
twitter.com/Profinit_EU

Facebook
facebook.com/Profinit.EU

Youtube
Profinit EU