

Úvod do Hadoop ekosystému

Martin Macák

Fakulta informatiky, Masarykova univerzita, Brno

14. 3. 2019

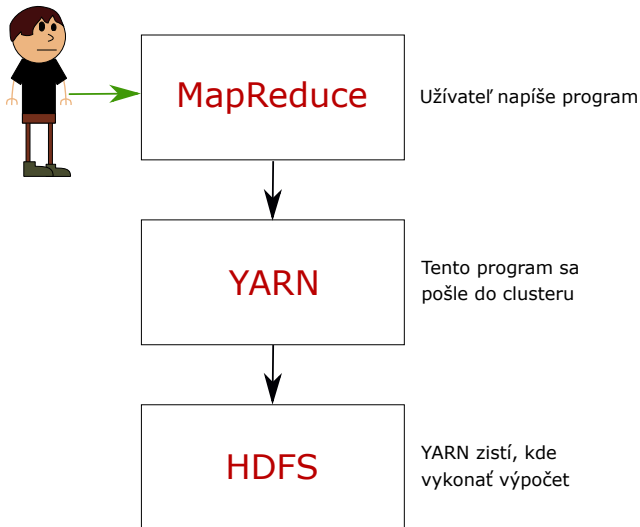
Osnova prednášky

1. Hadoop
2. Nástroje Hadoop ekosystému

Hadoop

- HDFS – distribuovaný file system
- MapReduce – framework na definovanie úlohy na spracovanie dát
- YARN – framework na vykonanie úlohy (kde? ako?)

Hadoop



Hadoop – možnosti inštalácie

- Standalone
 - defaultne
 - jeden stroj, jeden proces
 - super na debugging MapReduce programov
- Pseudo-Distributed
 - jeden stroj, viacej procesov
 - využíva HDFS a YARN
 - stále super iba na testovanie programov
- Fully Distributed
 - cluster
 - ťažšie sa konfiguruje
 - typicky produkčné prostredie

Hadoop – možnosti inštalácie

- Hadoop distribúcie
 - ľahšia inštalácia a konfigurácia
 - viacero nástrojov Hadoop ekosystému

cloudera[®]



Hadoop

- HDFS
- MapReduce
- YARN

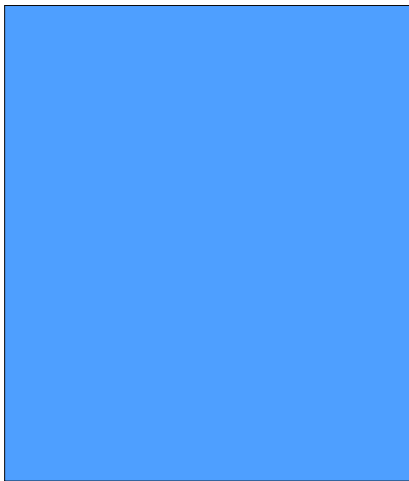
Hadoop

- **HDFS**
- MapReduce
- YARN

- cluster, na prácu s veľkými datasetmi
- dávkové spracovanie
- odolnosť proti chybám (výpadok stroja, ...)

- Data Nodes – obsahujú dáta
- Name Node – obsahuje info, kde tie dáta nájdeme

Velký súbtor



Veľký súbor



defaultne 128 MB veľkosť

aký efekt by mala nižšia veľkosť bloku?

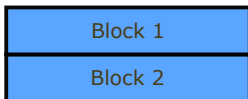
paralelizmus, overhead

aký efekt by mala vyššia veľkosť bloku?

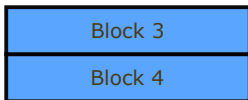
overhead, paralelizmus

HDFS

Data Node 1



Data Node 2



Data Node 3



Name node 1



- **ak crashne Data Node?**
- ak crashne Name Node?

HDFS – replikácia

Data Node 1

Block 1	Block 5
Block 2	Block 6

Data Node 2

Block 3	Block 1
Block 4	Block 2

Data Node 3

Block 5	Block 3
Block 6	Block 4

Name node 1

BigFile01: Block 1: DN 1

BigFile01: Block 2: DN 1

BigFile01: Block 3: DN 2

BigFile01: Block 4: DN 2

BigFile01: Block 5: DN 3

BigFile01: Block 6: DN 3

BigFile01: Block 1: DN 2

BigFile01: Block 2: DN 2

BigFile01: Block 3: DN 3

BigFile01: Block 4: DN 3

BigFile01: Block 5: DN 1

BigFile01: Block 6: DN 1

HDFS – replikácia

- defaultne 3 pre Fully Distributed mód, inak 1
- aký efekt by mal vyšší replikačný faktor?
spoľahlivosť, redundancia, efektívnosť zápisov
- aký efekt by mal nižší replikačný faktor?
redundancia, efektívnosť zápisov, spoľahlivosť

- ak crashne Data Node?
- **ak crashne Name Node?**

HDFS – strata Name Nodu

- bez Name Nodu nevieme, kde sú dáta
- čo sa dá robiť, aby sme z neho nestratili informácie?
- A) súbory s metadátami
 - **fsimage** (snapshot file systému pri štarte)
 - **edits** (edity vo file systéme)
 - dlho to trvá
- B) náhradný Name Node
 - treba každý daný časový interval synchronizovať s hlavným Name Nodom

- **hadoop fs**
- podobné Linuxu: **-ls**, **-mkdir**, **-touch**, **-cp**, **-mv**, ...
- z lokálneho file systému do HDFS: **hadoop fs -put**
- z HDFS do lokálneho file systému: **hadoop fs -get**
- **hadoop fs -help**

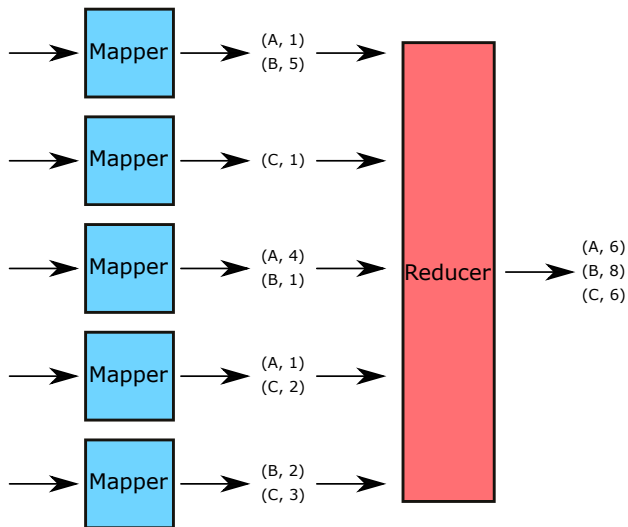
Hadoop

- HDFS
- **MapReduce**
- YARN

MapReduce

- programátor definuje **map** a **reduce**
- **map**
 - operácia, ktorá sa dá vykonať paralelne na časti datasetu
 - vracia key-value páry
- **reduce**
 - operácia, ktorá kombinuje výsledky **map** operácií

MapReduce



- treba sa zamyslieť nad dvomi otázkami:
 1. aké key-value páry majú vychádzať z Mapperov?
 2. ako skombinovať hodnoty s rovnakými kľúčmi?

MapReduce – Java demo

- trieda Mapper
 <**input key type, input value type, output key type, output value type**>
- trieda Reducer
 <**input key type, input value type, output key type, output value type**>
- main
 - trieda Job, ktorá používa Mapper a Reducer
 - pre objekt triedy Job treba nakonfigurovať niekoľko vlastností
 - následne sa môže spustiť

Hadoop

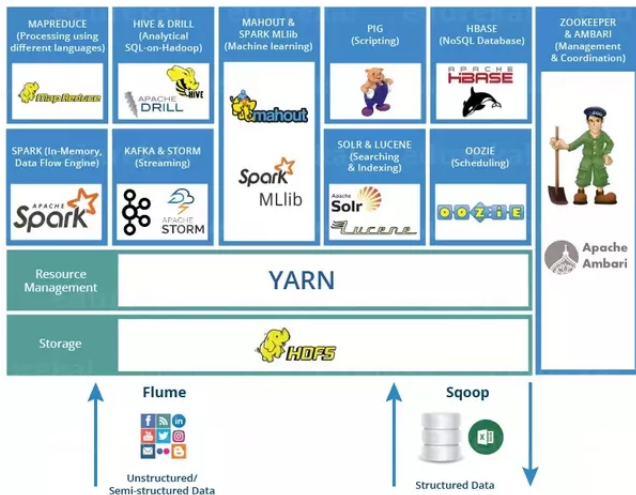
- HDFS
- MapReduce
- **YARN**

- koordinácia úloh v clustery
- drží si info o disku, RAM, CPU, ...
- na základe týchto informácií rozhoduje o priebehu danej úlohy
- taktiež rieši preplánovanie úloh pri výpadkoch

- Resource Manager
 - běží na Name Node, má na starosti celý cluster
- Node Manager
 - běží na Data Node, má na starosti len svoj uzol
 - každý Data Node má svoj Node Manager

Hadoop ekosystém

- súbor nástrojov, ktoré sa časom objavili okolo Hadoopu



- SQL nad MapReduce – HQL

```
SELECT word, count(1) AS count FROM
  (SELECT explode(split(cities, ' '))
   AS word FROM users)tempUsers
GROUP BY word
```

- manipulácie dát nad MapReduce – Pig Latin

```
input = LOAD '/users/cities.txt'  
      AS(sentence:Chararray);  
words = FOREACH input  
      GENERATE FLATTEN(TOKENIZE(line, ' '))  
grouped = GROUP words BY word;  
result = FOREACH grouped  
      GENERATE group, COUNT(words);
```

- column-family NoSQL databáza nad HDFS
- sám o sebe poskytuje len CRUD operácie
- na dotazovanie treba použiť iný nástroj (Hive, Drill, ...)

- SQL dotazovanie nad viacerými úložiskami
- podporuje napr. HDFS, HBase, MongoDB, ...

- machine learning algorithmy

Zookeeper

- koordinuje nástroje Hadoop ekosystému
- udržiava konfigurácie, poskytuje synchronizáciu, ...

1. Hadoop
 - HDFS, MapReduce, YARN
2. Hadoop ekosystém
 - v praxi treba vedieť viac nástrojov
 - Hive, Pig, HBase, Drill, ...



macak@mail.muni.cz