NÁSKOK
DÍKY
ZNALOSTEM

PROFINIT

NDBI047
Aplikace bigdat v Data Science

MapReduce

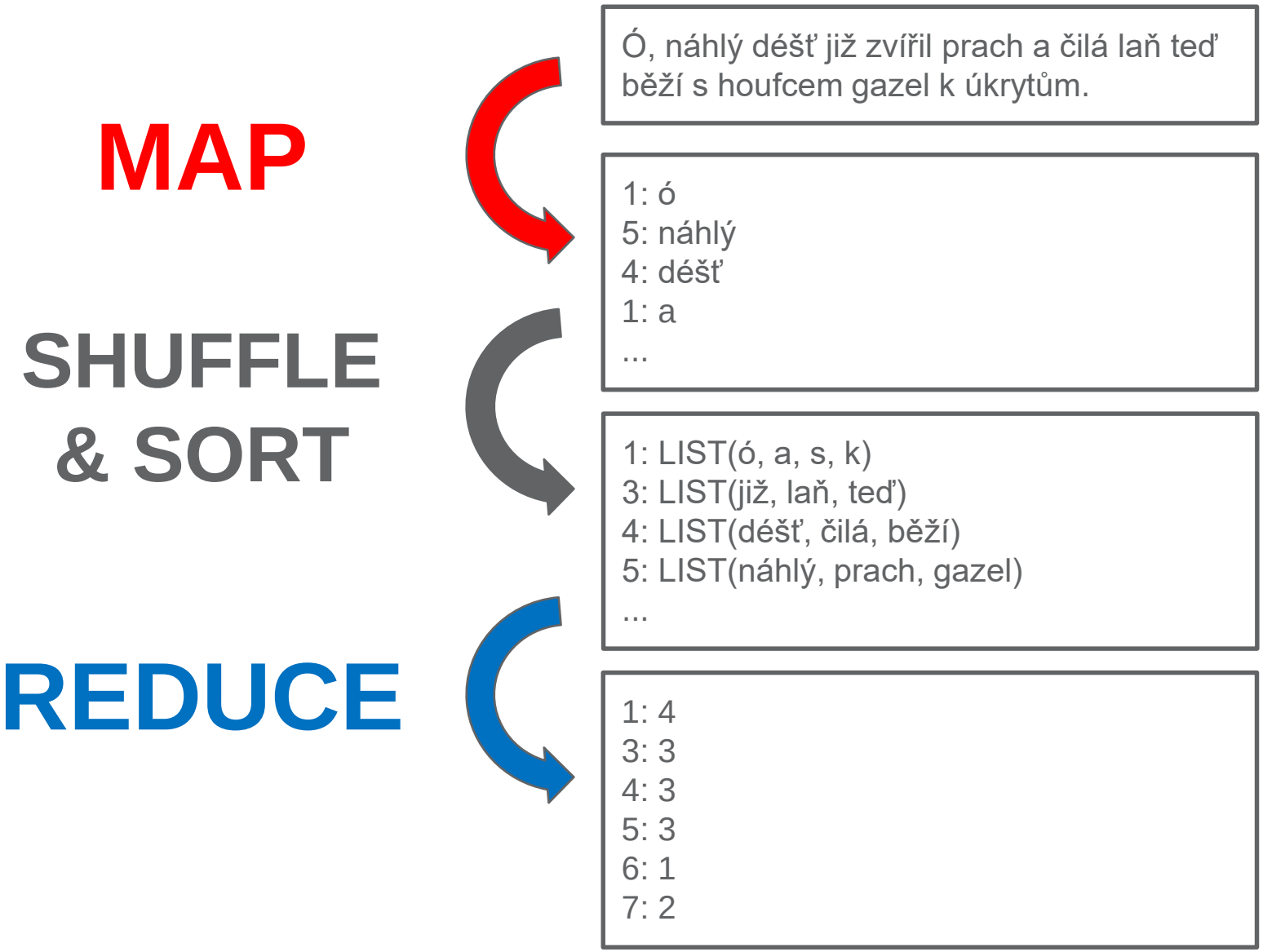
Milan Kratochvíl, Jan Hučín

8. 3. 2019

MapReduce

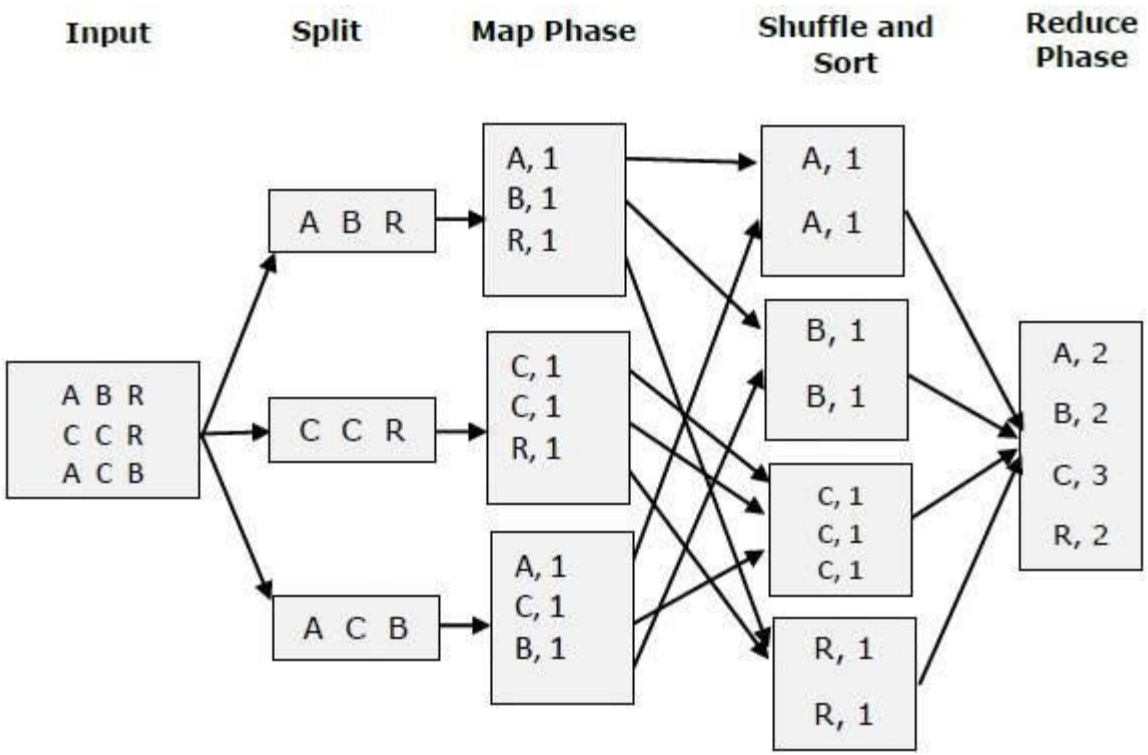
- Paradigma (programovací model) pro paralelní zpracování
- Cyklus MR:
  - načti data
  - transformuj data do párů <klíč, hodnota> – fáze **map**
  - proveď případný předvýpočet/agregaci – fáze **combine**
  - shromáždi páry se stejným klíčem – fáze **shuffle & sort**
  - proveď výpočet/agregaci odděleně pro každý klíč – fáze **reduce**
  - výsledek zapiš

MapReduce – příklad: Počet slov podle délky



PROFINIT

MapReduce – výpočty a tok dat



PROFINIT

Příprava

Split

- Pokud je soubor větší než 1 HDFS blok, splitne se na menší
- Lze zakázat splitování – pak každý mapper zpracuje celý soubor

Fyzický plán

- Podle rozmístění bloků – plán, co se kde bude počítat
  - Snaha zpracovat data tam, kde jsou uložena
  - Co nejméně transferů přes síť
  - Využití replikace dat v HDFS

Mapper

- Vstupní údaje **mapuje** (transformuje) na hodnoty typu <klíč, hodnota>
- Klíče i hodnoty mohou být cokoliv – třeba i složitá struktura
- Data na konci zpravidla setříděna podle klíče
- Mnoho paralelních jobů
- 1 split → 1 mapper
- **Výstup se zapisuje na lokální disk**

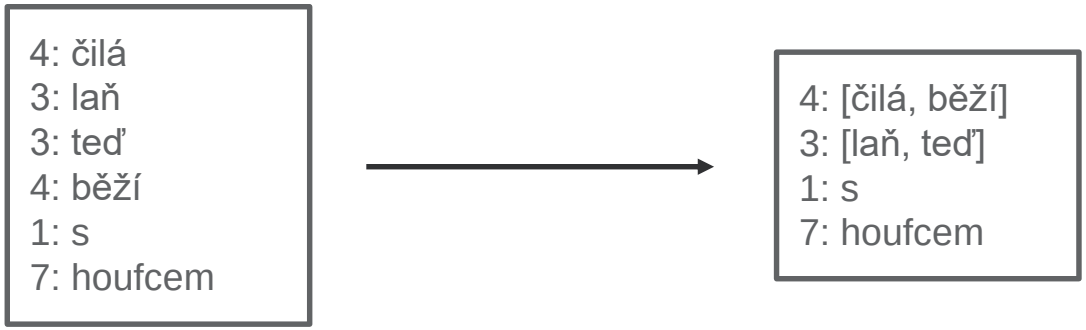
... čilá laň teď běží s houfcem...

4: čilá
3: laň
3: teď
4: běží
1: s
7: houfcem

PROFINIT

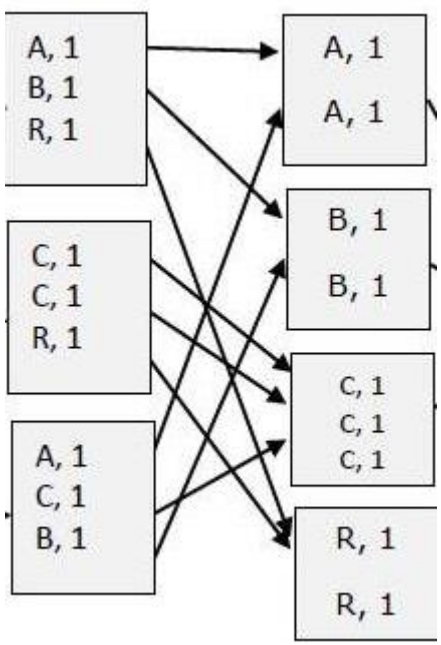
Combiner

- Volitelná část zpracování mapperu – agregace
  - dělá to, co by musel udělat reducer – šetří mu práci
- Proč použít combiner
  - data jsou načtena v paměti, ušetří se IO operace – čtení
  - často je výstup menší než vstup, ušetří se IO operace – zápis (a následné čtení)



Shuffle & Sort

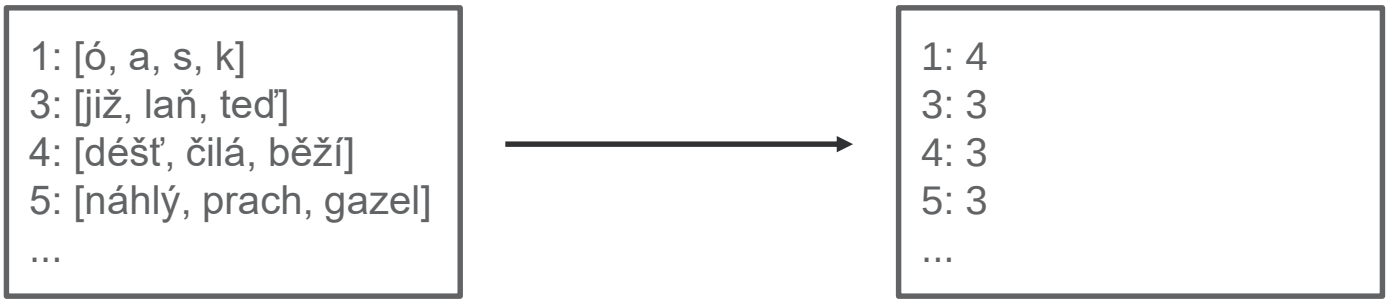
- Probíhá během/po skončení mapperů
- Vstup: **vygenerované soubory z mapperů**
- Všechna data se **stejným klíčem slije na jeden node**:
  - načtení na jednotlivých nodech (disková operace)
  - určení, co se kam pošle
  - přenos po síti
  - sloučení dat se stejným klíčem (merge)
- Optimalizace – malá data rovnou do reduceru, velká merguje na lokálním disku
- Typicky nejnáročnější operace



PROFINIT

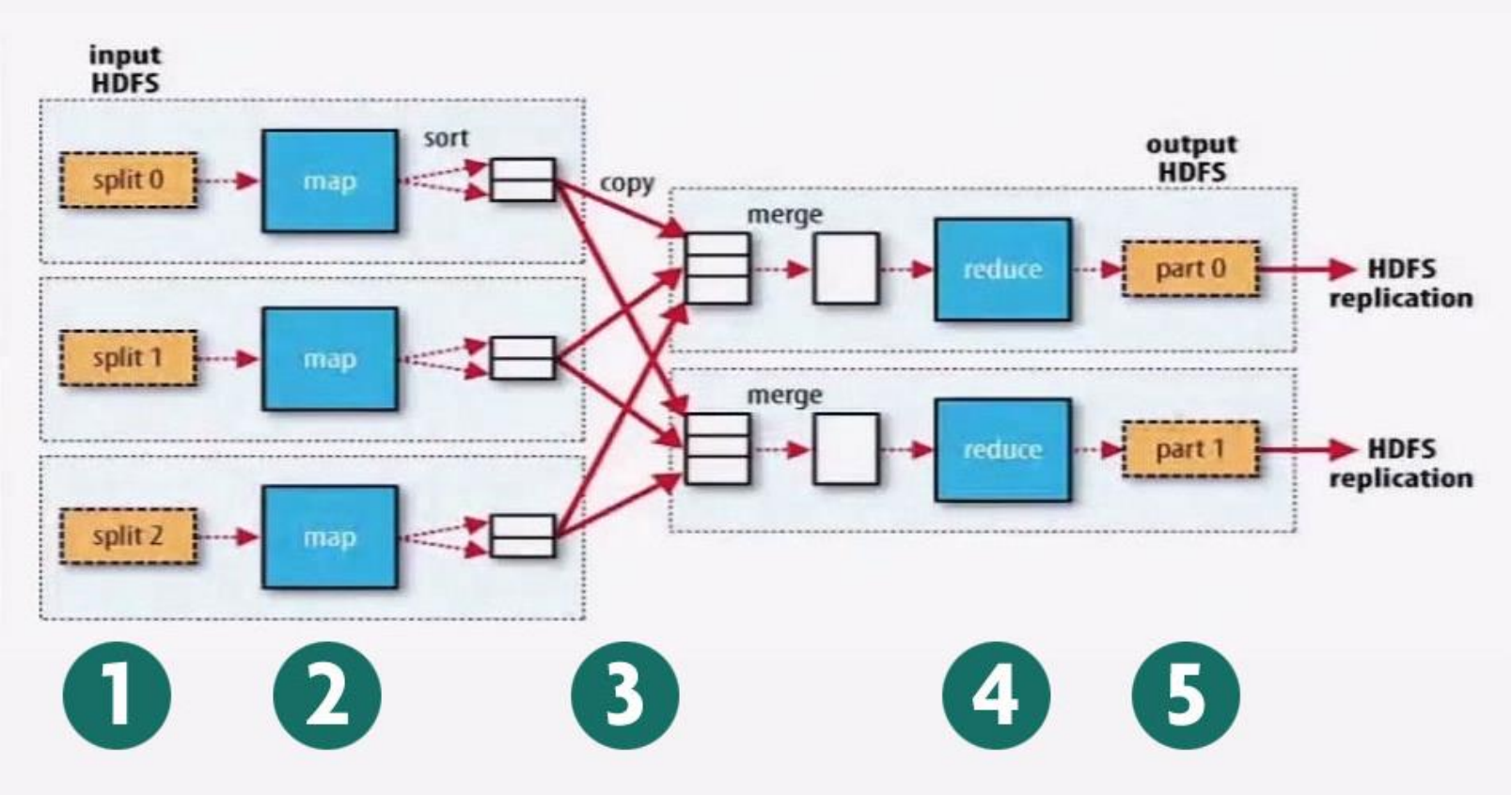
Reduce

- Čte data shromážděná pomocí Shuffle & Sort
- Agreguje data se stejným klíčem
- Počet reducerů lze definovat
- Každý reducer generuje 1 soubor do HDFS
- Výstup obecně nestříděný



PROFINIT

MapReduce – toky dat ještě jinak



https://content.pivotal.io/blog/hadoop-101-programming-mapreduce-with-native-libraries-hive-pig-and-cascading

PROFINIT

Třída úloh, které lze MapReduce zpracovat

Formální požadavky

- Každá reduce operace ▲ musí splňovat
  a) asociativitu, tj. (A ▲ B) ▲ C = A ▲ (B ▲ C)
  b) existenci neutrálního prvku ◊, tj. A ▲ ◊ = A
  c) komutativita, tj. (A ▲ B) = (B ▲ A)
- Proč
  a) neovlivníme pořadí vykonávaných operací
  b) node, který nemá výstup/nezpracovává data, neovlivní výsledek
  c) mapper/shuffle může změnit pořadí dat
     komutativita umožňuje použít combiner

PROFINIT

Příklady – vhodné úlohy

- „Školní úlohy“
  - počet slov v textu
  - četnost slov
- Praktičtější úlohy
  - reporting – načítání řady dílčích výsledků (prodeje)
    - podle klienta
    - produktu
    - lokality
  - řazení dat (sortování)
  - výpis pořadí např. prodávaných výrobků podle prodaných kusů
  - filtrování dat, validace
  - hledání unikátních hodnot (SELECT DISTINCT a FROM t)
  - extrakce snímků z videa

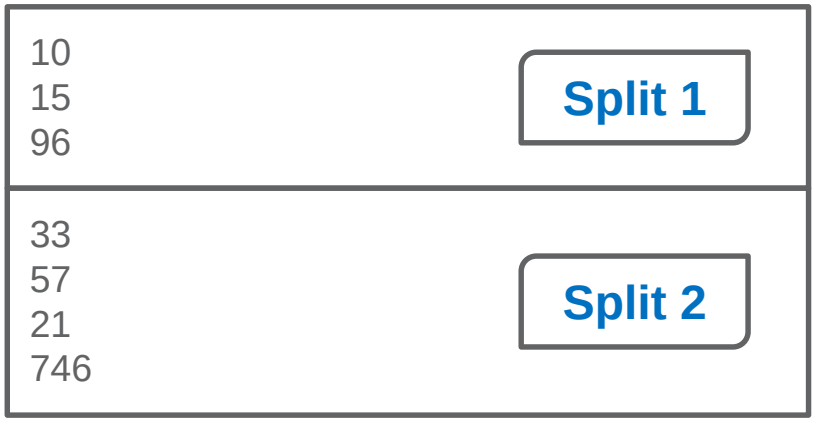
Příklady – nevhodné úlohy

- Medián
  - nesplňuje asociativitu
- Průměr
  - nesplňuje asociativitu
- Násobení matic
  - nesplňuje komutativitu

- ALE: **Úlohy lze vhodně přeformulovat!**

PROFINIT

Průměr



- Combiner
  - spočítat průměr
- Výstup mapperu
  - <SPLIT1, 40.3333>
  - <SPLIT2, 214.25>
- Výstup reduceru
  - 127.291666

- Combiner
  - spočítat průměr **a počet**
- Výstup mapperu
  - <SPLIT1, (40.3333, 3)>
  - <SPLIT2, (214.25, 4)>
- Výstup reduceru
  - AVG:139.714

PROFINIT

Příklady

„SQL“

- Hive jako tradiční dotazovací nástroj používá MapReduce. Jak?

- `SELECT COUNT(*) FROM SOME_TABLE;`

- MAP
  - pro každý řádek zapíšeme stejný klíč a číslo 1
  - `__row|1`
- REDUCE
  - jen jeden klíč!
  - na výstup zapíšeme počet (nebo sumu) hodnot ke klíči row
  - `__count|633214`

PROFINIT

„SQL“

```
SELECT COUNT(*), NAME FROM SOME_TABLE
WHERE COUNTRY='CZ'
GROUP BY NAME
HAVING COUNT(*) > 100;
```

- MAP
  - přeskočíme všechny záznamy, které nemají CZ
  - pro každý řádek zapíšeme stejný klíč – NAME
  - hodnota bude 1
  - ```
    Novak|1
    Sladek|1
    ```
- REDUCE
  - tolik klíčů, kolik máme jmen!
  - spočítáme ke každému klíči počet (nebo sumu) hodnot ke klíč
  - na výstup zapíšeme, jen ty, které jsou větší než 100
  - ```
    Novak|654
    Sladek|162
    ```

PROFINIT

Vlastnosti MapReduce

- Často nestačí jeden běh MapReduce → řetězení více běhů
  - výstup běhu → dočasné úložiště → vstup dalšího běhu
  - místo řetězení celých běhů lze někdy řetězit jen mappery
  - typicky filtry, validace
- Spolehlivé, když jiné nástroje selžou, MapReduce jede
- Dlouhá doba, než se zpracování spustí
  - desítky sekund až minuty
- Malé požadavky na operační paměť
  - v paměti se drží jen malé bloky
  - slučování fragmentů probíhá merge sortem na discích
- **Velmi četné zapisování na disk**
  - nicméně na HDFS je jenom finální vstup a výstup
  - generuje mnoho dočasných dat
- Tedy ve srovnání s jinými nástroji výrazně pomalejší

MapReduce – kdy ano, kdy ne

Kdy ano:

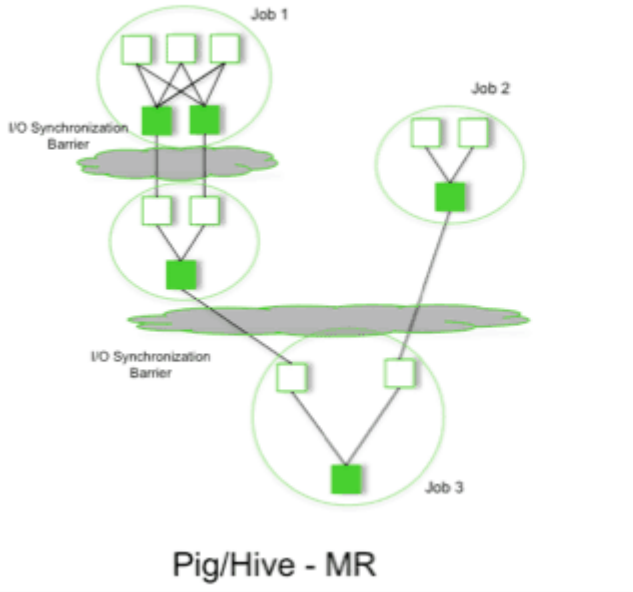
- Pokud úloha umožňuje paralelizaci
  - lze převést na úlohu vhodnou pro MapReduce
- Při práci se skutečně velkými daty (TB, PB)
  - tak velká data se v paměti nedají uchovávat/zpracovat
- Při práci s výpočetně náročnými úlohami, kde se spíše data načítají než zapisují
- Když není čas kritický
- Stačí dávkové zpracování
- Když není dostatek paměti v clusteru

Kdy ne:

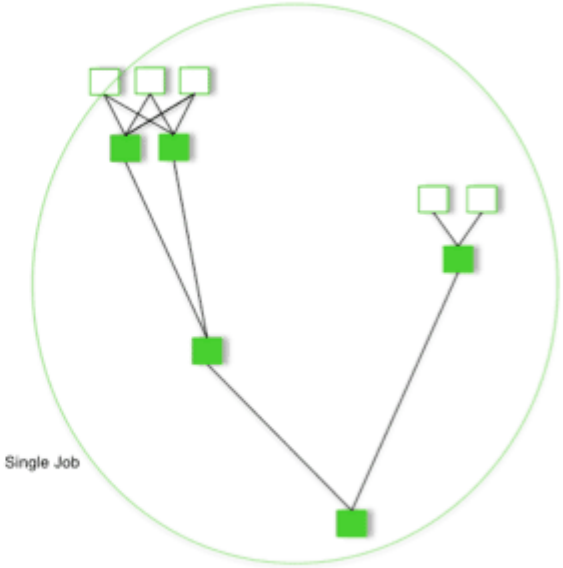
- opačné případy

Alternativy v Hadoop

- Apache Tez
  - optimalizovaná „verze“ MapReduce
  - princip je podobný, stále se zapisují mezivýsledky na disk
  - ale množství zápisů se optimalizuje, některé úkoly se slučují
  - násobně zvyšuje rychlost a prostupnost dat



Pig/Hive - MR



Pig/Hive - Tez

https://hortonworks.com/blog/introducing-tez-faster-hadoop-processing/

Alternativy v Hadoop

- Apache Spark
  - příští přednáška
  - podobné paradigma
  - místo ukládání mezivýsledků na disk vše drží v paměti
  - často řádové rozdíly ve výkonu
  - vhodný i pro ad hoc analýzy

- Přijďte příště a dozvíte se více!

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