


Databázové systémy

Map-Reduce

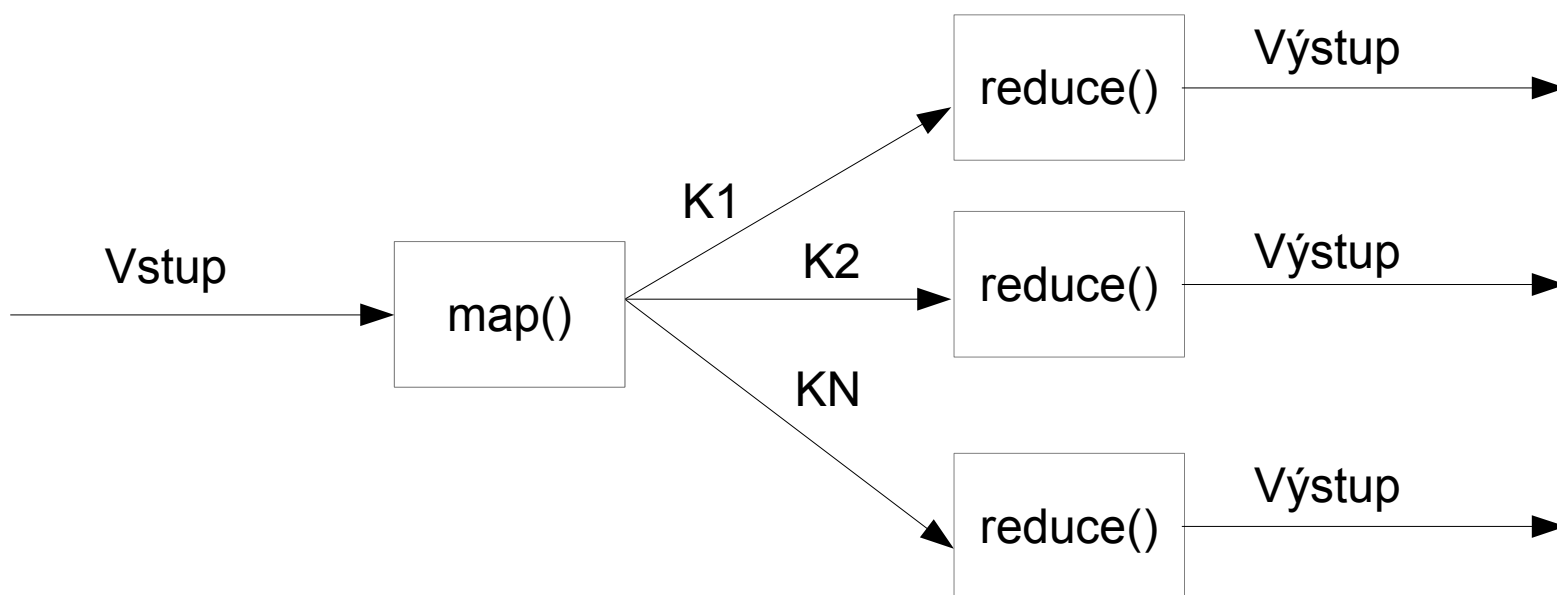
Motivácia

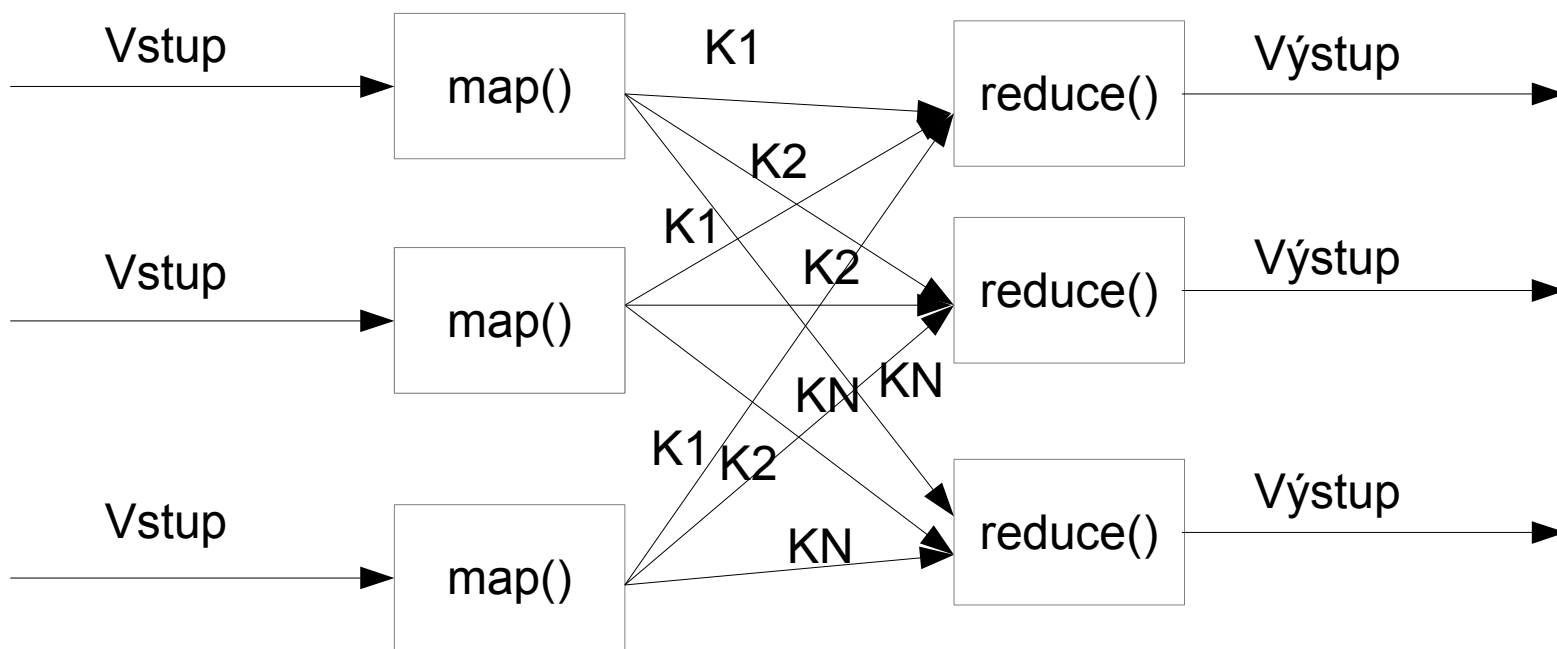
- Agregáty sú navrhnuté tak, aby bola pohodlná najčastejšia, typická práca s nimi
 - V objednávke máme zoznam objednaných produktov
- Čo v prípade, že manažér si chce pozrieť počty predajov jednotlivých produktov?
 - Musí prejsť všetky objednávky
- ...ale dáta máme v clustri, nedá sa to nejako využiť?

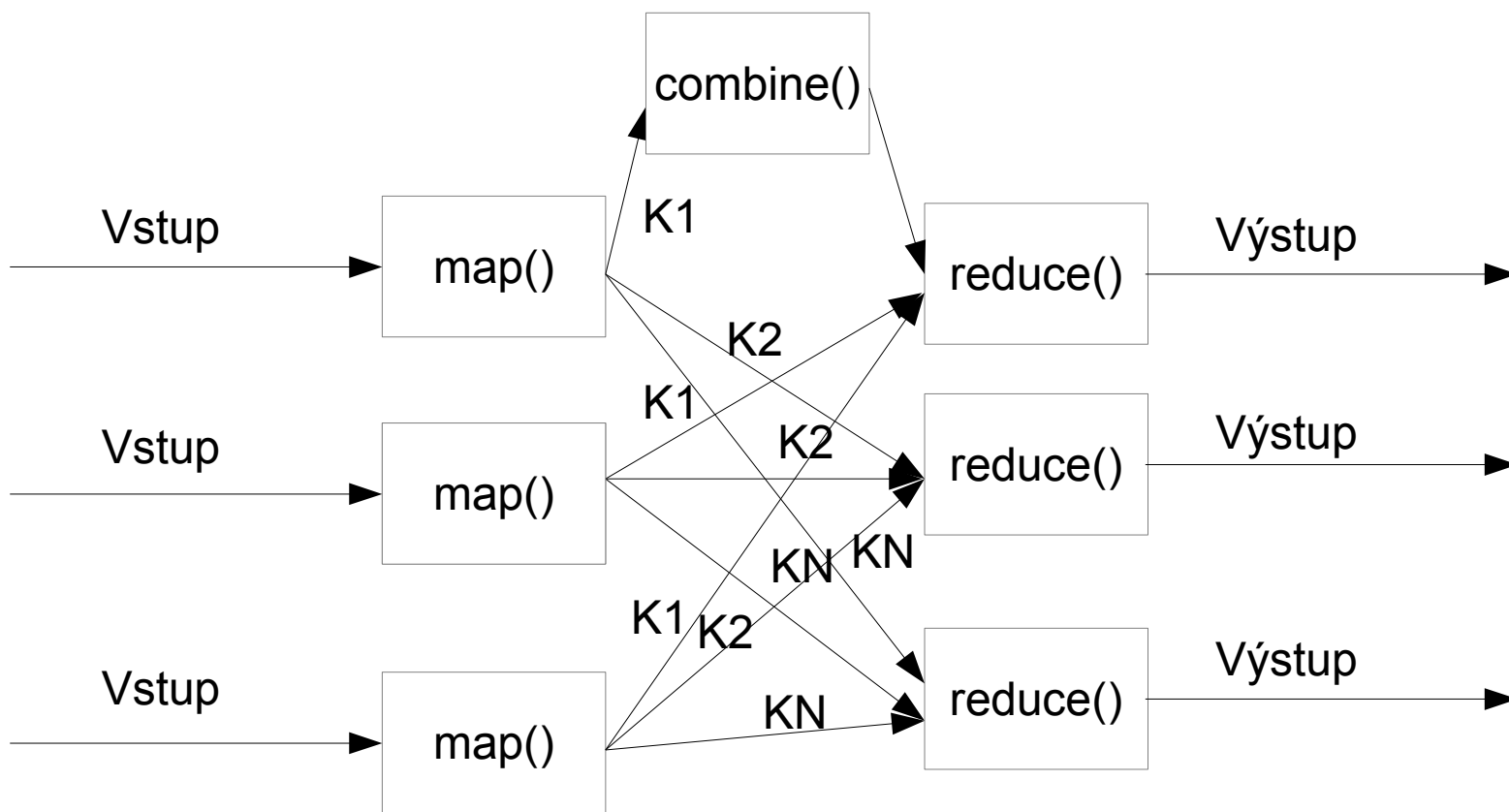
MapReduce framework

- Originál z Google, open source 
- žiaden dátový model, všetko v súboroch
 - GFS, resp. HDFS
- používateľ dodáva základné funkcie
 - map
 - reduce
 - combine
 - reader, writer
- framework sa postará o všetko ostatné

- map
 - $\text{map}(\text{item}) \rightarrow 0 \text{ a viac } \langle \text{Key}, \text{Value} \rangle \text{ párov}$
- reduce
 - $\text{reduce}(\text{key}, \text{list-of-values}) \rightarrow 0 \text{ a viac záznamov}$







Príklad - weblog

- CSV: UserID, URL, timestamp, additional-info
- Spočítaj všetky prístupy do domény (v URL)
- $\text{map}(\text{record}) \rightarrow \langle \text{domain}, \text{NULL} \rangle$
- $\text{reduce}(\text{domain}, \text{list of NULLs}) \rightarrow \langle \text{domain}, \text{count} \rangle$

Príklad - weblog

- CSV: UserID, URL, timestamp, additional-info
- Spočítaj všetky prístupy do domény (v URL)
- $\text{map}(\text{record}) \rightarrow \langle \text{domain}, \text{NULL} \rangle$
- $\text{combine}(\text{domain}, \text{list of NULLs}) \rightarrow \langle \text{domain}, \text{count} \rangle$
- $\text{reduce}(\text{domain}, \text{list of counts}) \rightarrow \langle \text{domain}, \text{sum} \rangle$

Ukážka

- wordcount na našom fakultnom SMART klastri

MapReduce

- žiadny dátový model, dáta v súboroch
- poskytneme len zopár metód (map, reduce)
- systém vykoná ostatné
 - fault-tolerant (nejaký uzol môže zomrieť)
 - škálovateľne (môžeme pridávať uzly)

MapReduce

- žiadny dátový model, dáta v súboroch
- poskytneme len zopár metód (map, reduce)
- systém vykoná ostatné
 - fault-tolerant (nejaký uzol môže zomrieť)
 - škálovateľne (môžeme pridávať uzly)
- predsa len je to veľa programovania
- chýba nám deklaratívnosť



Hive a Pig



- Hive – schéma, SQL-like rozhranie
 - ak viete SQL, tak viete aj Hive
- Pig – špeciálny jazyk (Pig Latin a Pig Commands) pre manipuláciu s dátami
- Obidva sa prekladajú do MapReduce jobov

Zhrnutie

- NoSQL – nie všetko je vhodné riešiť RDBMS
- flexibilnejšia schéma (nabudúce uvidíte viac)
- masívna škálovateľnosť
- veľká výkonnosť
- nižšia konzistencia
- absencia deklaratívneho dopytovania (nie je úplne pravda, veď máme Hive)