

MapReduce für Hauptspeicherdatenbanksysteme

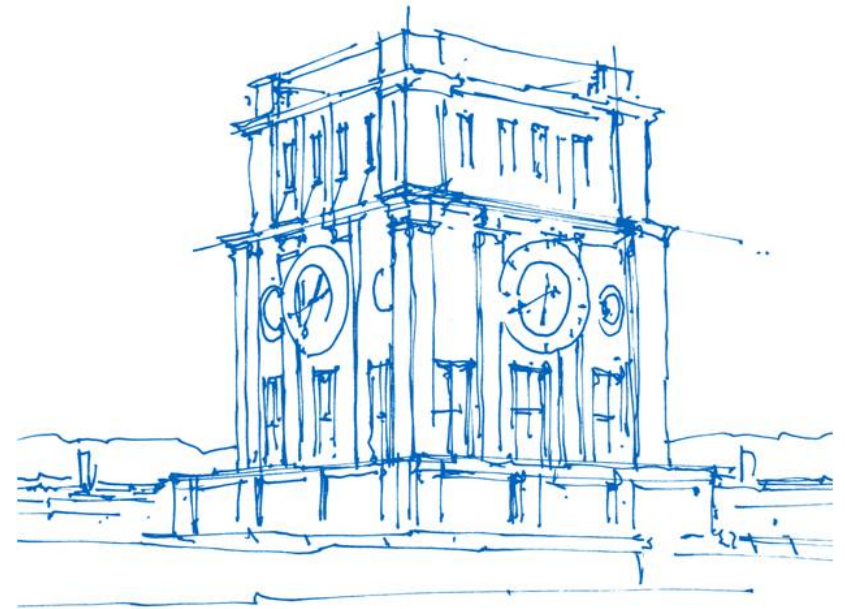
Michael Schwarz

Technische Universität München

Fakultät für Informatik

Lehrstuhl für Datenbanksysteme

06.02.2018



Uhrenturm der TUM

Übersicht

- MapReduce
- Relationale Algebra mit MapReduce
- Implementierung
- Evaluation

MapReduce

- Programmiermodell für die verteilte, parallele Berechnung mit großen Datenmengen
- Daten werden auf den Festplatten der Worker partitioniert gespeichert.
- Zwischenergebnisse werden auf den Festplatten der Worker gespeichert.

- Basierend auf drei Phasen:

- **Map**

$\{(k,v)\}^* \rightarrow \{(l,u)\}^*$

- **Shuffle**

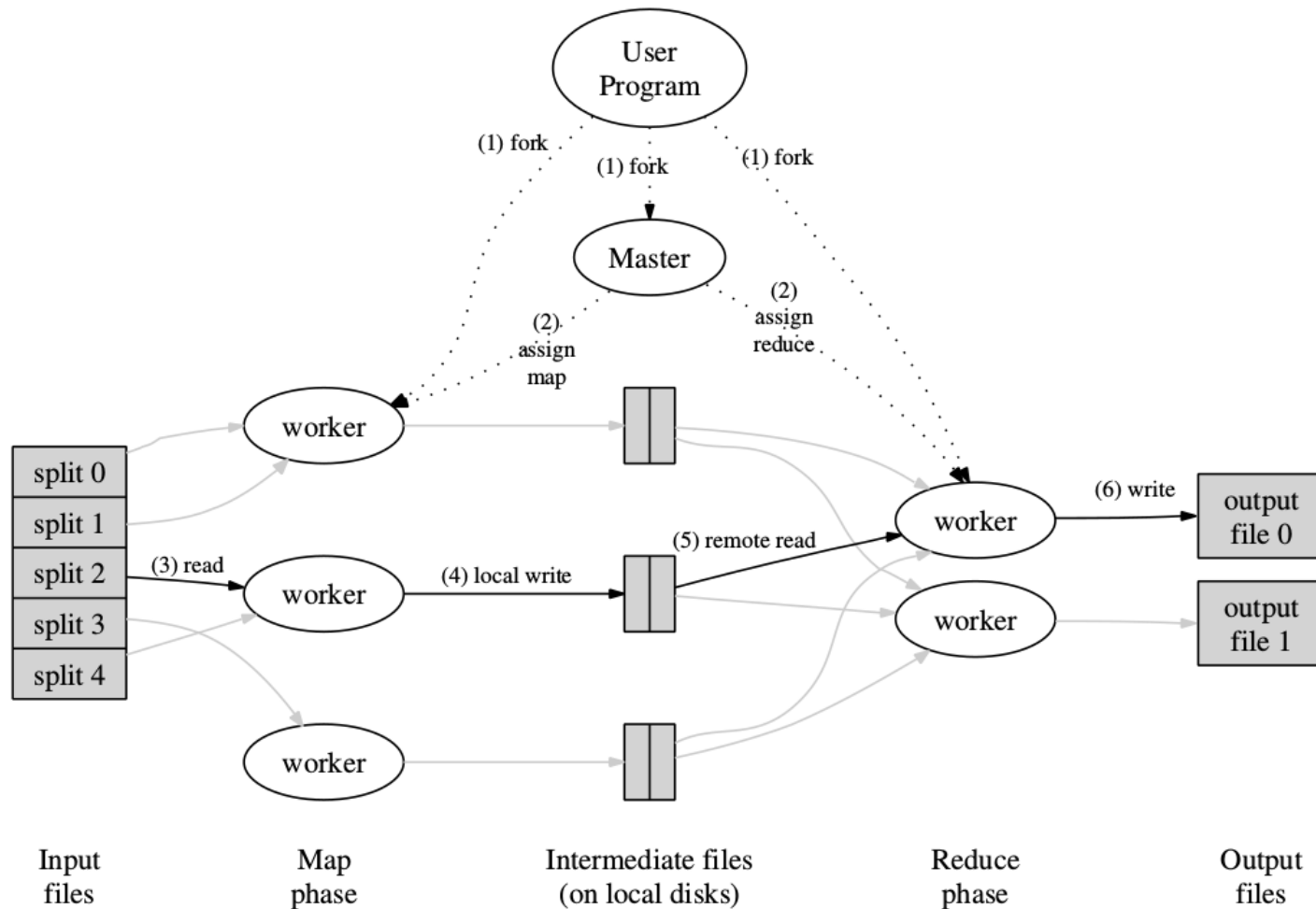
$\{(l,u)\}^* \rightarrow \{(l,\{u\}^*)\}^*$

- **Reduce**

$\{(l,\{u\}^*)\}^* \rightarrow \{(l,u)\}^*$

Nutzerspezifiziert

MapReduce



Quelle: Jeffrey Dean, Sanji Ghemawat: MapReduce: simplified data processing on large clusters

Relationale Algebra mit MapReduce

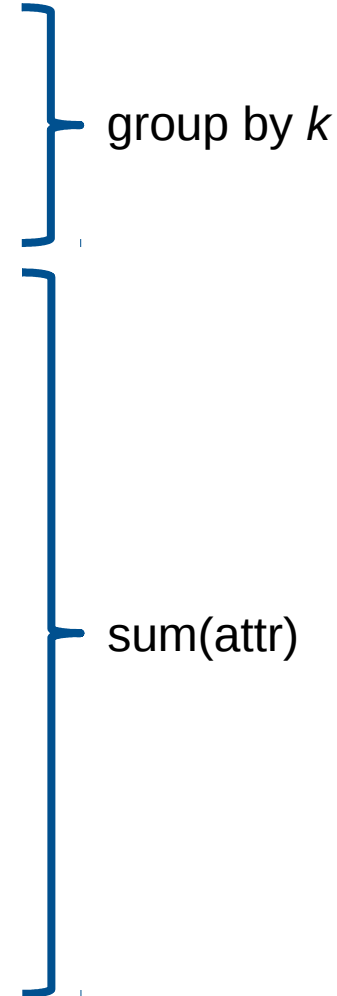
Operatoren:

- Selektion
- Projektion
- Aggregatsfunktionen
 - Summierung
 - Anzahl
 - Durchschnitt
 - Min
 - Max

sum() mit MapReduce

```
void map (int k, Record* v) {  
    emit((*v)[k],v);  
}
```

```
void reduce(int k, vector<Record*> v) {  
    int sum = 0;  
    for(vector<Record*> it = v.begin();  
        it != v.end(); ++it) {  
        sum += (*it)[attr];  
    }  
    collect(k,sum);  
}
```



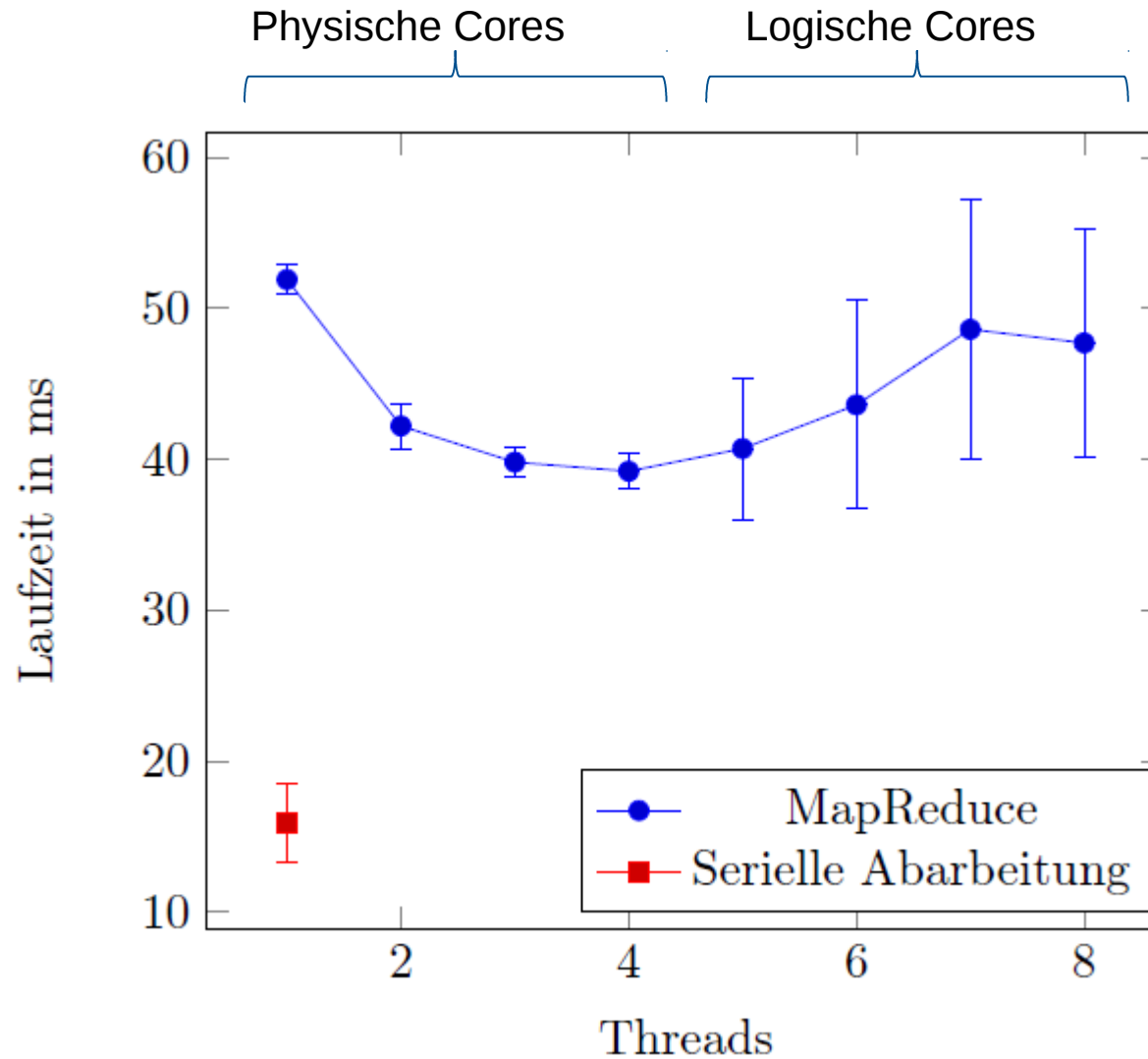
Evaluation

TestszENARIO:

- 1 Million Einträge mit je 10 Attributen
- Gruppierung nach Attribut a, Summierung von Attribut b
- Testläufe mit verschiedener Anzahl Threads
- Vergleichswerte:
Einfache, naive Implementierung mittels Hashmap

- Testumgebung:
 - Intel HM86 Express Chipset
 - Intel Core i7 4700HQ Prozessor
 - 16gb DDR3-SDRAM (1600MHz)

Evaluation



Zusammenfassung

- MapReduce ist für Hauptspeicherdatenbanksysteme implementierbar.
- MapReduce ist nur unter bestimmten Voraussetzungen schneller als herkömmliche Datenverarbeitungsstrategien.
- Einfache Definition der Map- und Reduce-Funktion durch Nutzer möglich.
- Aber: Mit SQL auch machbar.

Fragen?