

## Big Data Analytics in PArADISE

(Privacy AwaRe Assistive Distributed Information System Environment)

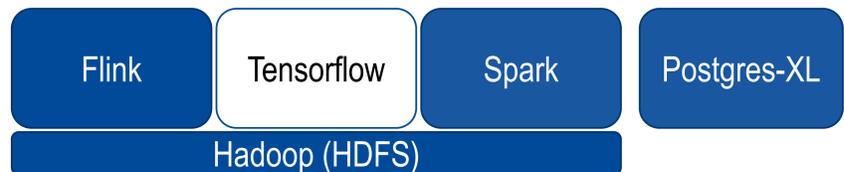
„MapReduce and parallel DBMSs: friends or foes?“

Projektarbeit SS 2017

- Stonebraker et al. publizierten 2010 die Ergebnisse eines Vergleichs zwischen Hadoop 0.19.0 und parallelen DBMS
- Cluster mit 100 Knoten
- Datensätze:
  - 1 TB Twitter-Follower-Graph mit 10 Mrd. Einträgen
  - 100 GB PageRank
  - 2 TB Weblog

- Nachstellung des Vergleichs mit Flink 1.2.0 und Spark 2.1.1 auf Hadoop 2.7.2 und Postgres-XL 9.5r1.4 als paralleles DBMS
- Cluster mit 3 Knoten
- Datensätze:
  - 26 GB Twitter-Follower-Graph
  - 1,15 GB PageRank
  - 4,29 GB Weblog

	Hadoop	DBMS-X	Vertica
Grep-Task	284 s	194 s	108 s
Weblog	1146 s	740 s	268 s
Join-Task	1158 s	32 s	55 s



• Resultat: Hadoop macht parallele DBMS nicht obsolet

- Tensorflow im Projektverlauf entfallen
- Resultat: Postgres-XL kann mit Flink und Spark mithalten

### Grep-Task

Twitter-Follower-Graph

ID1	ID2
12343454	86968792
29656457	94665834
37695979	81632765



- Suche nach einem Substring in einem großen Datensatz
- Keine Sortierung und keine Indexierung

Flink	Spark	Postgres-XL
100 s	53 s	121 s

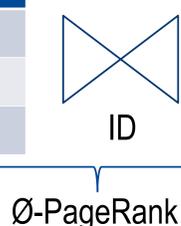
### Join-Task

PageRank

ID	PageRank
1	0,0016546
2	0,0857657
3	0,4534646

Weblog

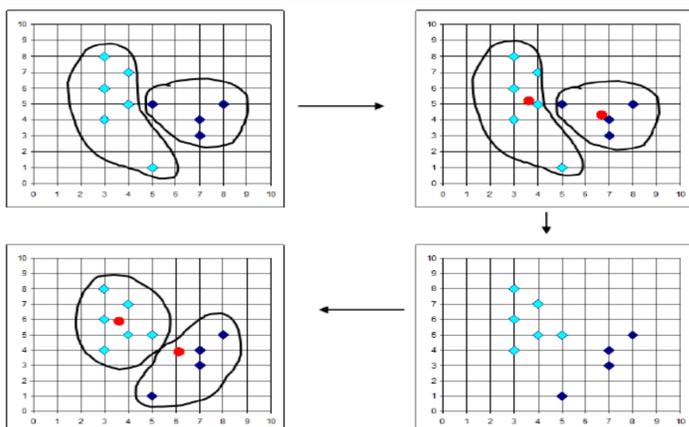
ID	IP	t
1	10	150
2	08	10
3	10	30



1. IP-Adresse ermitteln, die die meisten Twitter-Accounts in einem bestimmten Zeitraum besucht hat
  2. Join (PageRank ⋈ Weblog): Summe über PageRanks der von der IP-Adresse besuchten Twitter-Accounts bilden
  3. Aggregation: Durchschnitt über Summen ausgeben
- Bedingung: keine Sortierung

Flink	Spark	Postgres-XL
121 s	140 s	27 s

### k-Means



1. Centroide wählen
2. Distanzen berechnen
3. Daten den Centroiden zuordnen und somit Cluster bilden
4. Wiederholen bis sich die Cluster nicht mehr verändern oder Iterationsobergrenze erreicht

Flink	Spark	Postgres-XL
705 s	917 s	335 s